UNIWERSYTET WARMIŃSKO MAZURSKI W OLSZTYNIE
WYDZIAŁ MATEMATYKI I INFORMATYKI

Jakub Mikulski
Informatyka

Zastosowanie metod bioinformatycznych do analiz różnic w ekspresji genów.

Praca Inżynierska
Wykonana w Katedrze Botaniki i Ekologii Ewolucyjnej
pod kierunkiem dr. Łukasz Paukszto

Olsztyn 2026

UNIVERSITY OF WARMIA AND MAZURY IN OLSZTYN
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Jakub Mikulski
Computer Science

Bioinformatics methods for analyzing differences in gene expression.

Engineering Thesis
Prepared at the Department of Botany and Evolutionary Ecology
under the supervision of dr. Łukasz Paukszto

Olsztyn 2026

Spis treści

Table of content

Streszczenie

Celem pracy było stworzenie skryptu w języku R do analizy dużych zbiorów danych genetycznych. Projekt polegał na stworzeniu kompletnego potoku przetwarzania danych od ich wczytania, oczyszczenia, przez normalizację, aż po zaawansowaną analizę statystyczną i wizualizację. W ramach pracy wykorzystano środowisko RStudio oraz specjalistyczne biblioteki (m.in. edgeR, DESeq2, ggplot2). Zaimplementowano algorytmy uczenia maszynowego do redukcji wymiarowości (PCA, nMDS) oraz klastrowania, co pozwoliło na wykrycie ukrytych wzorców w danych. Głównym osiągnięciem projektu było stworzenie modelu, który z tysięcy badanych sekwencji automatycznie wyselekcjonował 19 kluczowych genów o zmiennej ekspresji. Przeprowadzona analiza pozwoliła na dalsze zawężenie wyników i wskazanie 3 genów o najsilniejszym trendzie zmian oraz 2 unikalnych genów, które reagują na badany czynnik w sposób specyficzny, niewystępujący w innych grupach. Potwierdza to skuteczność i precyzję napisanego kodu w zastosowaniach bioinformatycznych.

Abstract

The objective of this thesis was to develop an R script for the analysis of large-scale genetic datasets. The project involved designing a comprehensive data processing pipeline, ranging from data loading and cleaning, through normalization, to advanced statistical analysis and visualization. The study utilized the RStudio environment and specialized libraries (including edgeR, DESeq2, and ggplot2). Machine learning algorithms for dimensionality reduction (PCA, nMDS) and clustering were implemented, enabling the detection of latent patterns within the data. The main achievement of the project was the development of a model that automatically selected 19 key genes with differential expression from thousands of analyzed sequences. The conducted analysis allowed for further narrowing of the results to identify 3 genes with the strongest change trends and 2 unique genes that respond to the studied factor in a specific manner, distinct from other groups. This demonstrates the effectiveness and precision of the developed code in bioinformatics applications.

Wstęp

Współczesna diagnostyka molekularna w coraz większym stopniu opiera się na technologii sekwencjonowania RNA (RNA-seq), która umożliwia wgląd w dynamiczne zmiany profilu ekspresji genów pod wpływem czynników zewnętrznych. Jednym z kierunków badań w biologii medycznej jest rola pęcherzyków zewnątrzkomórkowych (EVs) w komunikacji międzykomórkowej. Pęcherzyki te, pełniąc rolę naturalnych nośników informacji molekularnej, mogą modulować fenotyp komórek biorców, co ma kluczowe znaczenie m.in. w progresji chorób nowotworowych i tworzeniu nisz przerzutowych. Niniejsza praca koncentruje się na bioinformatycznej analizie zmian transkryptomicznych zachodzących w komórkach poddanych działaniu EV, z uwzględnieniem dynamiki czasowej tych procesów, co pozwala na identyfikację kluczowych ścieżek sygnałowych aktywowanych w odpowiedzi na traktowanie.
W ramach niniejszej pracy przeprowadzono kompleksową analizę bioinformatyczną danych transkryptomicznych obejmujących 63 241 genów. Wykorzystanie technologii RNA-seq pozwoliło na precyzyjne monitorowanie zmian w ekspresji genów wywołanych działaniem pęcherzyków zewnątrzkomórkowych

RNA-seq

Sekwencjonowanie RNA, powszechnie określane jako RNA-seq, stanowi fundamentalną technologię we współczesnej biologii i naukach klinicznych, wykorzystującą metody wysokoprzepustowego sekwencjonowania do szczegółowego badania transkryptomu komórki. Transkryptom to zbiór cząsteczek RNA, które są niezbędne do interpretacji funkcjonalnych elementów genomu oraz zrozumienia procesów rozwojowych i mechanizmów chorobowych. Metoda ta pozwala badaczom nie tylko określić samą sekwencję nukleotydową cząsteczek RNA, ale przede wszystkim precyzyjnie oszacować ilości konkretnych gatunków RNA w danej populacji. W przeciwieństwie do wcześniejszych metod, takich jak mikro macierze, RNA-seq oferuje znacznie wyższą rozdzielczość i szerszy zakres dynamiczny, co pozwala na badanie dynamicznej natury transkrypcji bez konieczności posiadania wcześniejszej wiedzy o analizowanych sekwencjach.

Cel i zastosowanie metody RNA-seq

Głównym celem metody RNA-seq jest określenie sekwencji nukleotydowej cząsteczek RNA oraz ilościowe oznaczenie konkretnych gatunków RNA w badanej populacji cząsteczek. Technologia ta służy do uzyskania szczegółowego wglądu w transkryptom, co pozwala na

interpretację funkcjonalnych elementów genomu oraz zrozumienie procesów rozwojowych i mechanizmów powstawania chorób. [1]

Zastosowania metody RNA-seq obejmuje między innymi:

- Pomiar ekspresji genów i transkryptów: Pozwala określić, które geny są aktywne w danej próbce i jak silna jest ich ekspresja w różnych warunkach (np. w zdrowej tkance vs. tkance nowotworowej).
- Odkrywanie nowych struktur genetycznych: Metoda umożliwia wykrywanie nowych egzonów, całych transkryptów oraz genów, które nie zostały wcześniej opisane w bazach danych.
- Wykrywanie wariantów i mutacji: Technologia ta służy do identyfikacji biomarkerów, mutacji chorobotwórczych oraz zmian w regulacji genetycznej.
- Badania eQTL i sQTL: Integracja danych RNA-seq z danymi genotypowymi umożliwia identyfikację miejsc w genomie (loci), które wpływają na ilość lub sposób składania RNA.
- Zastosowania kliniczne i diagnostyczne: Metoda ta otwiera drogę do nowoczesnej diagnostyki, w tym monitorowania ciąży (badanie RNA płodu we krwi matki) czy wczesnego wykrywania nowotworów. [2]

Metoda RNA-seq jest dziś wykorzystywana we współczesnej medycynie co pozwala na skuteczne identyfikowanie zagrożeń związanych z nowotworami. Autorzy jednego z badań sprawdzili czy opracowana przez nich metoda RNA-seq sprawdzi się w przypadku odkrywania markerów nawrotu raka piersi. Dokonując analizy nową metodą RNA-seq 138 przypadków raka piersi wykryto znacznie więcej potencjalnych markerów (1307) powiązanych z ryzykiem nawrotu, a także setki markerów, których wcześniejsze technologie nie obejmowały. Dzięki RNA-seq udało się zidentyfikować całe sieci współ-ekspresjonujących się genów cyklu komórkowego związanych ze złym oraz lepszym rokowaniem. Wykonana praca pozwoliła na odkrycie nowych, złożonych sygnatur, które w przyszłości mogą stać się podstawą bardziej precyzyjnych testów prognozujących nawrót raka piersi i personalizacji leczenia. [3]

Charakterystyka obiektu badań

W dokonanym badaniu występują następujące czynniki:

CTR (Control) – jest to grupa kontrolna komórek hodowanych w identycznych warunkach bez dodatku czynnika EV. Grupa ta konieczna jest w celu dokonania rozróżnienia zmian wywołanych przez pęcherzyki od naturalnych procesów zachodzących w komórkach w czasie trwania hodowli. Grupa ta dzieli się na następujące okresy czasowe: 0 godzin (początek

hodowli) stanowi bazowy profil względem, którego interpretowane są zmiany zachodzące w kolejnych etapach eksperymentu, 12 godzin oraz 24 godziny od rozpoczęcia hodowli.
EV (Extracellular Vesicles) – grupa badawcza, składająca się z pęcherzyków zewnątrzkomórkowych komórek nowotworowych, które przenoszą ładunek molekularny. Są one naturalnymi nośnikami informacji. Grupa ta dzieli się na dwa okresy czasowe: 12 godzin oraz 24 godziny od potraktowania grupy CTR czynnikiem EV.

Dwie koncepcje

Obecny rozwój technologii oraz możliwości jakie dają systemy informatyczne pozwala na wnikliwe analizowanie danych bioinformatycznych poprzez różne narzędzia. Szybki rozwój mimo udzielania odpowiedzi na nurtujące pytania generuje nowe. Chociażby, której metody/koncepcji użyć w celu wykonania analizy, aby móc wyciągnąć odpowiednie wnioski?

1. Koncepcja „Jeden do jednego” (warunek vs kontrola)

Jest to klasyczne podejście analizy danych hodowlanych, które polega na porównaniu grupy EV z grupą CTR w jednym, konkretnym punkcie czasowym. [4] Zaletą takiego podejścia jest uzyskanie prostej odpowiedzi, które geny różnicują się w danym przedziale godzinowym. Metoda ta jest łatwa w interpretacji i wizualizacji. Choć model „jeden do jednego” cechuje się mniejszą złożonością obliczeniową, to dopiero podejście wieloczynnikowe umożliwia pełne wykorzystanie potencjału danych pochodzących z eksperymentów prowadzonych w czasie. [5] Wadą natomiast jest utrata informacji o dynamice zmienności. Zastosowanie tej metody uniemożliwia odpowiedzenie na pytanie czy w danym przedziale czasowym gen dopiero zwiększa swoją aktywność czy też jest aktywność zaczyna spadać. Poprzez to rozwiązanie tracimy możliwość uzyskania powiązań między punktami czasowymi.
Koncepcja ta została zastosowana w badaniu przeprowadzonej analizy – dotyczy „Analiza 1” dla grupy 12h oraz „Analiza 2” dla grupy 24h.

2. Koncepcja wieloczynnikowa

Koncepcja wieloczynnikowa jest zaawansowanym podejściem wywodzącym się z koncepcji poruszonej w poprzednim punkcie. Polega ona na modelowaniu ekspresji genów biorąc pod uwagę jednocześnie EV vs CTR oraz różne okresy czasowe. Zaletą takiego podejścia jest umożliwienie wykrycia genów, które reagują na EV inaczej w zależności od tego, ile czasu upłynęło. Takie zastosowanie pozwala na zidentyfikowanie genów wczesnej odpowiedzi i późnej odpowiedzi.
Koncepcja ta została zastosowana w badaniu przeprowadzonej analizy 3 – analizy wieloczynnikowej.

W niniejszej pracy zdecydowano się na zastosowanie modelu wieloczynnikowego, wykraczającego poza standardowe porównanie par grup. Podejście to umożliwia identyfikację genów wykazujących specyficzną dynamikę odpowiedzi na traktowanie pęcherzykami EV w czasie. Podejście to pozwala na oddzielenie genów reagujących na sam upływ czasu (procesy starzenia hodowli lub cyklu komórkowego) od genów, których ekspresja jest bezpośrednio i dynamicznie modulowana przez ładunek molekularny EV.

Metodyka oraz proces analizy (pipeline)

Środowisko pracy

W celu wykonania analizy RNA-seq wykorzystano środowisko programistyczne R studio przy użyciu specjalistycznych bibliotek bioinformatycznych takich jak „DESeq2", „edgeR", „biomaRt" co pozwoliło na realizację modeli prostych jak i wieloczynnikowych.

DESeq2 jest pakietem służącym do analizy różnicowej danych zliczanych (np. liczby odczytów na gen w RNA-seq) w celu wykrycia systematycznych zmian między różnymi warunkami eksperymentalnymi. Wykorzystuje on metody estymacji kurczenia (shrinkage estimation) dla dyspersji i zmian krotności (fold change), co poprawia stabilność i interpretowalność wyników, szczególnie w badaniach o małej liczbie powtórzeń. Pakiet opiera się na modelach liniowych (GLM) i rozkładzie ujemnym dwumianowym. [6]

EdgeR jest narzędziem przeznaczonym do badania różnicowej ekspresji danych zliczanych (np. RNA-seq). Podobnie do pakietu DESeq2, wykorzystuje model ujemnego dwumianu, aby uwzględnić zmienność biologiczną i techniczną. edgeR stosuje metody empirycznego Bayesa do moderowania stopnia dyspersji między genami, co umożliwia wiarygodne wnioskowanie statystyczne nawet przy minimalnej liczbie replikatów. [7]

BiomaRt to pakiet stanowiący interfejs API do usług internetowych BioMart (np. bazy Ensembl), który umożliwia integrację zbiorów danych genomowych. Służy głównie do mapowania identyfikatorów (np. zamiany identyfikatorów genów na identyfikatory białek lub transkryptów) oraz pobierania adnotacji dotyczących sekwencji, szlaków metabolicznych czy powiązań z chorobami bezpośrednio do środowiska R. [8]

Etap I: Przygotowanie środowiska oraz danych

Proces analizy rozpoczęto od załadowania kluczowych bibliotek w środowisku R, takich jak: DESeq2, edgeR (analiza różnicowa), biomaRt (anotacja genów) oraz ggplot2 (wizualizacja danych). Następnie zaimportowano surową macierz zliczeń z pliku counts_EV.csv.

```
library(DESeq2)
library(biomaRt)
library(edgeR)
library(gplots)
library(ggplot2)
```

```
count_table <- read.csv2("counts_EV.csv", row.names = 1)
```

Preprocesing i transformacja danych do formatu zgodnego z wymaganiami silników statystycznych w celu dalszych analiz. Zdefiniowanie struktury próbek w zmiennej pheno_data, przypisując 12 próbkom status kontrolny (CTR) oraz 8 próbkom status badawczy (EV) wraz z odpowiednimi punktami czasowymi. Nadanie kolejności wierszy według danych z kolumny pierwszej, usunięcie kolumny pierwszej. Zmiana nazw kolumn na „condition" oraz „time". Konwersja „condition" oraz „time" na typ faktorowy w celu ustawienia etykiet grup, które mają być porównywalne statystycznie oraz spełnienie wymagań dla pakietu DESeq2, aby analiza mogła zostać przeprowadzona poprawnie. Dodatnie kolumny „treatment" z kombinacją typu próbki oraz czasu, zmiana typu danych na faktor. Utworzenie zmiennej interakcji treatment pozwoliło na bezpośrednie porównania między konkretnymi grupami w modelach liniowych pakietu DESeq2

```
pheno_data <- data.frame(colnames(count_table), rep(c("CTR","EV"),c(12,8)),
c(0,12,24,0,12,24,0,12,24,0,12,24,12,24,12,24,12,24,12,24)) #utworzenie
zmiennej pheno_data (12 wierszy CTR i 8 wierszy EV, rozdzielenie czasu w
trzeciej kolumnie)
rownames(pheno_data) <- pheno_data[,1] #numerowanie wierszy według danych z
kolumny pierwszej
pheno_data <- pheno_data[,-1] #usunięcie kolumny pierwszej
colnames(pheno_data) <- c("condition", "time") #zmiana nazw pozostałych kolumn
na condition i time
pheno_data$condition <- as.factor(pheno_data$condition) #konwersja na faktor
pheno_data$time <- as.factor(pheno_data$time) #konwersja na faktor
pheno_data$treatment <- paste0(pheno_data$condition,"_", pheno_data$time)
#dodanie kolumny z kombinacją typu i czasu (czy jest to CTR czy EV i czas)
pheno_data$treatment <- as.factor(pheno_data$treatment) #konwersja na faktor
```

Kolejnym etapem preprocesingu było wyodrębnienie z głównego zbioru danych dwóch niezależnych podzbiorów odpowiadających punktom czasowym 12 i 24 godziny. Operacja ta była podyktowana koniecznością przeprowadzenia szczegółowych analiz porównawczych (EV vs. CTR) w ramach konkretnych interwałów czasowych.

Logika przekształceń obejmowała:

- Filtrację metadanych: Na podstawie kolumny time utworzono nowe ramki danych pheno_data_12 oraz pheno_data_24, zawierające informacje wyłącznie o próbkach z danego punktu czasowego.
- Normalizację struktury ramek: Dokonano reorganizacji kolumn w podzbiorach metadanych, przypisując nazwy próbek do pierwszej kolumny oraz ujednolicając nazewnictwo zmiennych (kolumny sample i treatment).
- Ekstrakcję macierzy zliczeń: Wykorzystując przefiltrowane metadane, dokonano selekcji odpowiednich kolumn z głównej tabeli count_table. W efekcie uzyskano dwie dedykowane macierze zliczeń: count_table_12 oraz count_table_24, które posłużyły jako dane wejściowe dla obiektów klasy DESeqDataSet.

```
####Utworzenie zmiennych dla próbek 12 i 24h oraz przypisanie im danych####
pheno_data_12 <- pheno_data[pheno_data$time == 12,]
pheno_data_24 <- pheno_data[pheno_data$time == 24,]

####Obrobienie zawartości pheno_data 12 i 24####
pheno_data_12[,2] <- pheno_data_12[,1] #zastąpienie kolumny drugiej pierwszą
pheno_data_12[,1] <- rownames(pheno_data_12)
pheno_data_12 <- pheno_data_12[,-3] #usunięcie kolumny trzeciej
colnames(pheno_data_12) <- c("sample","treatment") #zmiana nazw kolumn

pheno_data_24[,2] <- pheno_data_24[,1] #zastąpienie kolumny drugiej pierwszą
pheno_data_24[,1] <- rownames(pheno_data_24)
pheno_data_24 <- pheno_data_24[,-3]
colnames(pheno_data_24) <- c("sample","treatment")

#####Utworzenie tabel genów dla 12 i 24 godzin####
count_table_12 <- count_table[,colnames(count_table) %in%
rownames(pheno_data_12)]
count_table_24 <- count_table[,colnames(count_table) %in%
rownames(pheno_data_24)]
```

Nawiązanie połączenia z bazą danych Ensembl za pośrednictwem pakietu biomaRt. Ze względu na to, że surowe dane z sekwencjonowania identyfikowane są za pomocą kodów Ensembl ID, niezbędne było przeprowadzenie adnotacji biologicznej. Zastosowano funkcję getBM do pobrania kompleksowych informacji o genach gatunku Homo sapiens, obejmujących:

- oficjalne symbole genów (external_gene_name),
- opisy funkcji biologicznych (description),
- biotypy genów (gene_biotype).

Baza adnotacyjna została następnie zintegrowana (poprzez operację merge) z wynikami każdej z przeprowadzonych analiz różnicowych. Pozwoliło to na przekształcenie surowych wyników

statystycznych w czytelne zestawienia biologiczne, co było kluczowe dla identyfikacji genów o największym znaczeniu dla przebiegu eksperymentu, takich jak TCF4 czy RASD1.”

```
#####Nawiązanie połączenia z bazą danych ensembl. Wybranie danych dot.
człowieka i wskazanie konkretnych atrybutów.####
ensemblHs = useMart(host="ensembl.org", biomart="ENSEMBL_MART_ENSEMBL",
dataset="hsapiens_gene_ensembl")
allgenes.Ensembl = getBM(attributes=c("ensembl_gene_id", "external_gene_name",
"gene_biotype", "entrezgene_id", "description",
"entrezgene_accession"),mart=ensemblHs) # tabela z wybranymi atrybutami genów
```

Prawidłowe zdefiniowanie metadanych oraz synchronizacja macierzy zliczeń z opisem próbek stanowi fundament całej analizy statystycznej. Przeprowadzone w tym kroku transformacje (faktoryzacja zmiennych, segmentacja czasowa) pozwoliły na przygotowanie danych w formacie zgodnym z rygorystycznymi wymogami pakietów DESeq2 oraz edgeR. Dzięki temu wyeliminowano błędy związane z niepoprawnym przypisaniem grup referencyjnych na dalszych etapach pracy.

Etap II: Eksploracyjna analiza danych i kontrola jakości

Po wstępnym przygotowaniu macierzy zliczeń oraz metadanych, kolejnym niezbędnym krokiem w potoku analizy była eksploracyjna analiza danych. Głównym celem tego etapu było zweryfikowanie spójności biologicznej replikatów, identyfikacja ewentualnych próbek odstających (outliers) oraz ocena globalnego wpływu czynników doświadczalnych (traktowania EV i czasu) na profil transkryptomiczny. W ramach tego etapu wykonano następujące operacje:

- Normalizacja danych - zastosowano metodę TMM (Trimmed Mean of M-values) za pomocą funkcji calcNormFactors, aby zniwelować różnice wynikające z różnej głębokości sekwencjonowania bibliotek.
- Transformacja logarytmiczna - przeliczono surowe zliczenia na wartości CPM (Counts Per Million) w skali logarytmicznej, co pozwoliło na stabilizację wariancji i umożliwiło stosowanie metod opartych na dystansie euklidesowym.
- Analiza głównych składowych (PCA) - wykorzystano algorytm redukcji wymiarowości do wizualizacji podobieństwa między próbkami w przestrzeni dwuwymiarowej.
- Niemetryczne skalowanie wielowymiarowe (nMDS) - jako alternatywną metodę wizualizacji dystansu między próbkami zastosowano nMDS, co pozwoliło na potwierdzenie trendów zaobserwowanych w analizie PCA.

```
#####Przystosowanie próbek. Wygładzenie oraz ujednolicenie.
```

```
pheno_data$grouping <- paste(pheno_data$treatment, pheno_data$time, sep = ".")
Group <- factor(paste(pheno_data$time, pheno_data$treatment, sep = "_"))
y <- DGEList(counts = count_table, remove.zeros = TRUE) #Usunięcie zer
y <- calcNormFactors(y) #Normalizacja próbek
df_log <- cpm(y, log = TRUE, prior.count = 2) #cpm count per milion
dds.pcoa = pcoa(vegdist(t(df_log <- cpm(y, log = TRUE, prior.count = 2)),
                        method = "euclidean") / 1000)
scores <- dds.pcoa$vectors
percent <- dds.pcoa$values$Eigenvalues
cumulative_percent_variance <- (percent / sum( percent)) * 100
```

Analiza PCA

Kluczowym wyzwaniem w analizie danych pochodzących z sekwencjonowania RNA-seq jest ich wysoka wymiarowość – każda próbka opisana jest przez dziesiątki tysięcy zmiennych (genów). Aby umożliwić wizualną ocenę struktury danych oraz wykryć globalne zależności między próbkami, zastosowano Analizę Głównych Składowych (ang. Principal Component Analysis – PCA).

PCA jest matematyczną techniką redukcji wymiarowości, która przekształca zbiór skorelowanych zmiennych w mniejszą liczbę nieskorelowanych zmiennych, zwanych głównymi składowymi (PC). Pierwsza składowa (PC1) wyjaśnia największą część zmienności w danych, natomiast każda kolejna składowa tłumaczy pozostałą wariancję w stopniu malejącym. W bioinformatyce metoda ta pozwala na szybką identyfikację skupień próbek (klastrów) oraz wykrycie ewentualnych błędów technicznych (np. próbek odstających) [9].

```
#Wykonanie PCA (analiza głównych składowych)
df_log_transposed <- t(df_log) #Próbki w wierszach, geny w kolumnach
pca_results <- prcomp(df_log_transposed, scale. = TRUE)

#Przygotowanie danych do ggplot2
pca_scores <- as.data.frame(pca_results$x)
pca_scores$Sample <- rownames(pca_scores)

#Połączenie wyników PCA z danymi pheno_data
pca_scores <- merge(pca_scores, pheno_data, by.x = "Sample", by.y =
"row.names")

#Obliczenie procentu wariancji wyjaśnianej przez każdą składową
percent_variance <- round((pca_results$sdev^2 / sum(pca_results$sdev^2)) *
100, 1)

#Generowanie wykresu PCA
pca_plot <- ggplot(pca_scores, aes(x = PC1, y = PC2,
  color = condition,
  shape = time)) +
  geom_point(size = 4, alpha = 0.8) +
  scale_shape_manual(values = c("0" = 16, "12" = 17, "24" = 15)) +
#Zdefiniowane kształty: Kółko, Trójkąt, Kwadrat
  scale_color_manual(values = c("CTR" = "chartreuse4", "EV" =
"darkturquoise")) +
```

```
  #Etykiety osi z procentem wyjaśnionej wariancji
  xlab(paste0("PC1 (", percent_variance[1], "%)")) +
  ylab(paste0("PC2 (", percent_variance[2], "%)")) +
  #Etykiety punktów
  geom_text_repel(aes(label = Sample), size = 3, color = "black", max.overlaps
= 20) +
  ggtitle("PCA próbek (Normalizowane logCPM)") +
  coord_fixed(ratio = 1) +
  theme_bw() +
  theme(legend.title = element_text(face = "bold"))

print(pca_plot)

dev.copy(png, "PCA_Analiza_Eksploracyjna.png", width=700, height=500) #zapis
do pliku
dev.off() #zamknięcie okna
```

Analiza nMDS - Niemetryczne skalowanie wielowymiarowe

Jako alternatywną i uzupełniającą metodę wizualizacji podobieństwa próbek zastosowano niemetryczne skalowanie wielowymiarowe (ang. non-metric Multidimensional Scaling – nMDS). W przeciwieństwie do PCA, metoda nMDS nie opiera się na wyjaśnianiu wariancji poprzez liniowe kombinacje zmiennych, lecz na zachowaniu rangowych odległości (podobieństw) między próbkami w zredukowanej, dwuwymiarowej przestrzeni. Pozwala to na bardziej elastyczne oddanie dystansów biologicznych, które nie zawsze mają charakter liniowy. Do obliczenia macierzy dystansu wykorzystano miarę euklidesową na danych znormalizowanych logarytmicznie, a algorytm iteracyjnie poszukiwał optymalnego rozmieszczenia próbek na płaszczyźnie.

```
set.seed(42) #Ustawienie ziarna dla powtarzalności wyników nMDS
nmds_results <- metaMDS(df_log_transposed,
  distance = "euclidean",
  k = 2, #2 wymiary (X i Y)
  trymax = 100) #Zwiększenie prób

print(paste("nMDS Stress Value:", nmds_results$stress))

#Przygotowanie danych do ggplot2
#Wyciągnięcie współrzędnych dla próbek (sites)
nmds_scores <- as.data.frame(scores(nmds_results, display = "sites"))
nmds_scores$Sample <- rownames(nmds_scores)

#Połączenie wyników nMDS z danymi fenotypowymi (pheno_data)
nmds_scores <- merge(nmds_scores, pheno_data, by.x = "Sample", by.y =
"row.names")

#Generowanie wykresu nMDS
nmds_plot <- ggplot(nmds_scores, aes(x = NMDS1, y = NMDS2)) +
  #Elipsy liczone tylko na podstawie grupy condition
```

```
  stat_ellipse(aes(group = condition, color = condition),
    type = "t", linetype = "dashed", alpha = 0.5) +
  #Punkty z zachowaniem kolorów i kształtów
  geom_point(aes(color = condition, shape = time), size = 4, alpha = 0.8) +
  coord_fixed() +
  ggtitle("nMDS próbek (Normalizowane logCPM)") +
  scale_shape_manual(values = c("0" = 16, "12" = 17, "24" = 15)) +
  scale_color_manual(values = c("CTR" = "chartreuse4", "EV" =
"darkturquoise")) +
  geom_text_repel(aes(label = Sample), size = 3, color = "black", max.overlaps
= 20) +
  theme_bw()
```

W celu zweryfikowania, w jakim stopniu upływ czasu (0h, 12h, 24h) wpływa na ogólną zmienność danych transkryptomicznych, wygenerowano wykres nMDS z nałożonymi elipsami ufności dla poszczególnych punktów czasowych (Rysunek 3).

```
nmds_plot_czas <- ggplot(nmds_scores, aes(x = NMDS1, y = NMDS2, color = time))
+
  geom_point(size = 4) +
  stat_ellipse(aes(fill = time), geom = "polygon", alpha = 0.1) + #Elipsy dla
grup czasowych
  coord_fixed() +
  theme_bw() +
  ggtitle("Grupowanie próbek względem czasu (0h, 12h, 24h)")
```

Klastrowanie hierarchiczne

W celu weryfikacji struktury danych przeprowadzono klastrowanie hierarchiczne próbek. Macierz podobieństwa wyznaczono w oparciu o dystans euklidesowy, a do budowy dendrogramu wykorzystano metodę Warda. Podejście to pozwoliło na minimalizację zmienności wewnątrz grup, co uwidoczniło wyraźną separację próbek kontrolnych od badanych traktowanych EV.

```
#Obliczenie macierzy odległości
dist_matrix <- dist(df_log_transposed, method = "euclidean")

#Wykonanie klastrowania
hc_results <- hclust(dist_matrix, method = "ward.D2")

#Wyświetlenie dendrogramu
klaster <- plot(hc_results,
  main = "Hierarchiczne klastrowanie próbek (logCPM)",
  sub = "",
  xlab = "",
  labels = pheno_data$treatment,
  ylab = "Wysokość (Dystans euklidesowy)")
```

Zgodnie z dobrymi praktykami analizy danych opisywanymi przez Goldmeiera, proces badawczy rozpoczęto od eksploracyjnej analizy danych (Etap II). Pozwoliło to na wstępne zrozumienie struktury zbioru i wykrycie ewentualnych anomalii przed przystąpieniem do testowania hipotez statystycznych [10]. W tym celu zastosowano trzy niezależne metody wizualizacji (PCA, nMDS oraz klastrowanie hierarchiczne) umożliwiło uzyskanie spójnego obrazu struktury danych i potwierdziło wysoką jakość biologicznego materiału.

Etap III: Analiza różnicowej ekspresji genów (DGE)

Analiza 1 i analiza 2 - Porównanie warunków EV i CTR w poszczególnych punktach czasowych

Głównym celem tego etapu było zidentyfikowanie konkretnych genów, których poziom ekspresji uległ istotnej zmianie pod wpływem traktowania pęcherzykami zewnątrzkomórkowymi (EV) w porównaniu do grupy kontrolnej (CTR). Zastosowano tutaj koncepcję „jeden do jednego", przeprowadzając dwie niezależne analizy dla poszczególnych punktów czasowych: Analizę 1 (12h) oraz Analizę 2 (24h).

Do wyznaczenia różnic wykorzystano pakiet DESeq2, który implementuje zaawansowane modele statystyczne oparte na rozkładzie ujemnym dwumianowym. Proces ten obejmował następujące kroki:

- Definicja grup: Za pomocą funkcji relevel wskazano grupę CTR jako referencyjną, co umożliwiło poprawną interpretację kierunku zmian (zwiększona/zmniejszona ekspresja w EV względem CTR).
- Filtrowanie niskiej ekspresji: Usunięto geny o zerowej lub znikomej liczbie odczytów (rowSums > 1), co zwiększyło moc statystyczną testów.
- Kryteria istotności: Przyjęto rygorystyczne progi odcięcia: skorygowaną wartość p (padj < 0,05) oraz bezwzględną wartość zmiany krotności (|log2FoldChange| > 1).

```
#Utworzenie DESeq2 dla 12h
dds_12 <- DESeqDataSetFromMatrix(countData = count_table_12,
  colData = pheno_data_12,
  design = ~ treatment)

#Filtrowanie i analiza
dds_12$treatment <- relevel(dds_12$treatment, ref = "CTR")
dds_12 <- dds_12[rowSums(counts(dds_12)) > 1, ]
dds_12 <- DESeq(dds_12)

#Wyniki porównania EV vs CTR dla 12h
res_12 <- results(dds_12, contrast = c("treatment", "EV", "CTR"))
res_12 <- as.data.frame(res_12)
```

```
res_12 <- res_12[!is.na(res_12$padj), ]

#Oznaczenie zmian
res_12$change <- "Bez zmian"
res_12$change <- ifelse(res_12$log2FoldChange > 1 & res_12$padj < 0.05,
"Zwiększona ekspresja", res_12$change)
res_12$change <- ifelse(res_12$log2FoldChange < -1 & res_12$padj < 0.05,
"Zmniejszona ekspresja", res_12$change)

#Przygotowanie danych dla 24h
pheno_data_24_filtered <- pheno_data_24
count_table_24_filtered <- count_table_24

#Utworzenie obiektu DESeq2 dla 24h
dds_24 <- DESeqDataSetFromMatrix(countData = count_table_24_filtered,
  colData = pheno_data_24_filtered, design = ~ treatment)
#Filtrowanie i analiza
dds_24$treatment <- relevel(dds_24$treatment, ref = "CTR")
dds_24 <- dds_24[rowSums(counts(dds_24)) > 1, ]
dds_24 <- DESeq(dds_24)

#Wyniki porównania EV vs CTR dla 24h
res_24 <- results(dds_24, contrast = c("treatment", "EV", "CTR"))
res_24 <- as.data.frame(res_24)
res_24 <- res_24[!is.na(res_24$padj), ]

#Oznaczenie zmian
res_24$change <- "Bez znaczącej zmiany"
res_24$change <- ifelse(res_24$log2FoldChange > 1 & res_24$padj < 0.05,
"Zwiększona ekspresja", res_24$change)
res_24$change <- ifelse(res_24$log2FoldChange < -1 & res_24$padj < 0.05,
"Zmniejszona ekspresja", res_24$change)
```

Analiza 3 Model wieloczynnikowy i analiza interakcji

Ostatnim etapem analizy różnicowej było zastosowanie modelu wieloczynnikowego, który pozwolił na wyjście poza statyczne porównania punktowe. Celem tej analizy było zidentyfikowanie genów, których odpowiedź na pęcherzyki zewnątrzkomórkowe (EV) zmienia się w sposób istotny wraz z upływem czasu. Etap ten stanowi najbardziej zaawansowany i kluczowy element dotychczasowej pracy. Wykorzystuje on modelowanie wieloczynnikowe do zbadania interakcji między traktowaniem (EV vs CTR) a czasem trwania eksperymentu.
Preprocessing danych, w celu wyodrębnienia jedynie tego co jest potrzebne do konkretnego porównania następuje:

- Filtrowanie – wybrano tylko próbki z grupy EV i CTR oraz dwa punkty czasowe (12h i 24h).

- Synchronizacja – dokonano dopasowania tabeli zliczeń genów do tabeli opisowej. Jeżeli próbka nie przeszła pomyślnie kontroli jej jakości lub jej kryteria są nieodpowiednie, zostaje usunięta z obu tabel jednocześnie.
- Naprawa poziomów – usunięcie danych, których już nie ma w celu uniknięcia zaburzenia obliczeń statystycznych.

```
#Przygotowanie danych
pheno_data_analiza_3 <- pheno_data[
  (pheno_data$condition %in% c("EV", "CTR")) &
    (pheno_data$time %in% c(12, 24)), ]
pheno_data_analiza_3 <- droplevels(pheno_data_analiza_3)

count_table_analiza_3 <- count_table[, rownames(pheno_data_analiza_3)]
count_table_analiza_3 <- as.matrix(count_table_analiza_3)
mode(count_table_analiza_3) <- "numeric"
```

Kolejny etap analizy bioinformatycznej obejmował zdefiniowanie struktury statystycznej eksperymentu oraz normalizację danych w celu identyfikacji genów o różnicowej ekspresji.
W pierwszej kolejnośći skategoryzowano zmienne objaśniające poprzez przekształcenie ich w typ factor. Przy użyciu funkcji relevel ustalono poziomy odniesienia. Jako grupę bazową dla warunku „condition" przyjęto kontrolę (CTR) a dla czasu (time) punkt 12h.
Dane zliczeń genów wraz z metadanymi zostały zintegrowane w obiekcie strukturalnym DGEList. Podczas tego procesu usunięto geny o zerowej ekspresji we wszystkich próbkach (remove.zeros = TRUE), co zredukowało obciążenie obliczeniowe i poprawiło moc testową. Następnie przeprowadzono normalizację metodą TMM (Trimmed Mean of M-values) przy użyciu funkcji calcNormFactors.
W celu zbadania wpływu obu czynników jednocześnie, skonstruowano macierz projektu (design matrix) w oparciu o model liniowy z członem interakcji: condition * time. Model ten uwzględnia: efekt główny warunku, efekt główny czasu oraz, co najistotniejsze, interakcję między nimi. Zgodnie z podejściem prezentowanym przez Gutmana i Goldmeiera [11], kluczowe w analizie danych nie jest jedynie stosowanie algorytmów, ale zrozumienie modelu statystycznego, który pozwala odróżnić sygnał od szumu. W tym przypadku modelowanie interakcji pozwala na identyfikację genów, których profil odpowiedzi na czynnik EV zmienia się w sposób specyficzny wraz z upływem czasu, co jest kluczowe dla zrozumienia dynamiki procesów biologicznych. Ostatnim krokiem przygotowawczym było oszacowanie zmienności biologicznej za pomocą funkcji estimateDisp z parametrem robust = TRUE. Zastosowanie procedury empirycznego Bayesowskiego modelowania pozwoliło na stabilizację genów o niskiej ekspresji i ograniczenie wpływu wartości odstających.

```
pheno_data_analiza_3$condition <- as.factor(pheno_data_analiza_3$condition)
pheno_data_analiza_3$condition <- relevel(pheno_data_analiza_3$condition, ref
= "CTR")
pheno_data_analiza_3$time <- as.factor(pheno_data_analiza_3$time)
pheno_data_analiza_3$time <- relevel(pheno_data_analiza_3$time, ref = "12")
pheno_data_analiza_3 <- droplevels(pheno_data_analiza_3)

#Obiekt DGEList i normalizacja
dge_analiza_3 <- DGEList(counts = count_table_analiza_3, remove.zeros = TRUE)
dge_analiza_3 <- calcNormFactors(dge_analiza_3)

#Macierz modelu dwuczynnikowego
design_analiza_3 <- model.matrix(~ condition * time, data =
pheno_data_analiza_3)
print(colnames(design_analiza_3))

#Estymacja dyspersji i fitting GLM
dge_analiza_3 <- estimateDisp(dge_analiza_3, design_analiza_3, robust = TRUE)
fit_analiza_3 <- glmQLFit(dge_analiza_3, design_analiza_3)
```

Statystyczne poszukiwanie genów i przygotowanie do wykresów

W tym kroku sprawdzono, które geny reagują na badanie, a następnie dane ułożono tak, aby wynikowa heatmapa była czytelna.

1. Testowanie -Analiza różnicowa

Użycie modelu statystycznego w celu przeprowadzenia trzech niezależnych weryfikacji statystycznych:

- Czy leczenie działa? (coef = 2) Wyszukanie genów, które różnią się między grupą EV a kontrolą (CTR), bez brania pod uwagę czynnika czasowego.
- Czy czas ma znaczenie? (coef = 3) Wyszukanie genów, które same z siebie zmieniają się między 12 a 24 godziną trwania hodowli.
- Interakcja (coef = 4) Wyszukanie genów, które reagują na czynnik EV inaczej w 12 godzinie, a inaczej w 24 godzinie trwania hodowli. Pozwala to na wykrycie genów, których reakcja zależy od czasu trwania badania.

2. Wybór „najciekawszych” genów

Wyciągnięcie z interakcji tylko tych genów, które są statystycznie pewne poprzez zastosowanie filtra p-value < 0.01, co oznacza bardzo małe ryzyko błędu. Użycie poprawki Benjamini-Hochberga w celu kontroli współczynnika fałszywych odkryć. Jest to konieczne, ponieważ przy badaniu tysięcy genów istnieje ryzyko otrzymania przypadkowych wyników. Wykorzystanie tej metody pozwala na odrzucenie takich przypadków i pozostawienie tylko genów, które naprawdę się zmieniają.

3. Przeliczenie skali (Logarytmowanie)

Dane zliczeń genów zostały przeliczone na jednostki logCPM. W surowych danych jeden gen może mieć wartość 10, a inny 100 000 co może doprowadzić do całkowitego zakrycia genów o małych wartościach. Zastosowanie skali logarytmicznej wygładza te różnice dzięki czemu zmiany genów będą widoczne na heatmapie mimo zróżnicowanych wartości.

4. Układanie danych w celu ukazania ich na grafie

Ostatnia część kodu odpowiada za uporządkowanie tabeli, tak aby heatmapa była czytelna. W celu utworzenia nowych nazw etykiet połączono dane tak aby tworzyły nazwy „CTR_12h", „EV_24h" itd. Przeprowadzono ustawienie próbek tak aby na wykresie widoczne były one w następującej kombinacji CTR_12H -> EV_12H -> CTR_24H -> EV_24H.
Następnie w celu utworzenia legendy przygotowano tabelę (annotation_col), która na gotowym wykresie (Rysunek 9) tworzy kolorowe paski informujące odpowiadające przedziałowi czasowemu oraz grupie próbek.

Etap ten pozwolił na wyodrębnienie genów najsilniej reagujących na interakcję EV z czasem, a następnie uporządkowano i przeskalowano dane tak, aby wynikowa heatmapa była czytelna i ukazywała logiczny proces zmian.

```
#Test efektu głównego condition (EV vs CTR)
test_treatment_analiza_3 <- glmQLFTest(fit_analiza_3, coef = 2)

#Test efektu głównego time (24h vs 12h)
test_time_analiza_3 <- glmQLFTest(fit_analiza_3, coef = 3)

#Test interakcji condition × time
test_interaction_analiza_3 <- glmQLFTest(fit_analiza_3, coef = 4)

#Wyciągnięcie top genów z interakcji
top_interaction_analiza_3 <- topTags(test_interaction_analiza_3, n = Inf,
adjust.method = "BH", p.value = 0.01)
genes_interaction_analiza_3 <- rownames(top_interaction_analiza_3$table)

full_logCPM_analiza_3 <- cpm(dge_analiza_3, log = TRUE)
logCPM_analiza_3_filtered <-
full_logCPM_analiza_3[genes_interaction_analiza_3, ]

#Tworzenie kolumny kombinacji Grupa_Czas
pheno_data_analiza_3$comb_sort <- paste0(pheno_data_analiza_3$condition, "_",
pheno_data_analiza_3$time, "h")
#Definicja pożądanej kolejności poziomów
set_order <- c("CTR_12h", "EV_12h", "CTR_24h", "EV_24h")
#Zmiana kolejności poziomów na faktorze
pheno_data_analiza_3$comb_sort <- factor(pheno_data_analiza_3$comb_sort,
  levels = set_order)
#utalenie kolejnosci probek oraz sortownie
kolejnosc_probek <- order(pheno_data_analiza_3$comb_sort,
  rownames(pheno_data_analiza_3))
```

```
#Wyodrębnienie nazw próbek w kolejności
kolejnosc_probek <- rownames(pheno_data_analiza_3)[kolejnosc_probek]
#Sortowanie macierzy ekspresji genów (kolumn)
logCPM_sorted_analiza_3 <- logCPM_analiza_3_filtered[, kolejnosc_probek]
#Sortowanie macierzy adnotacji (wierszy)
annotation_col <- pheno_data_analiza_3[kolejnosc_probek, c("condition",
"time")]
colnames(annotation_col) <- c("Grupa", "Czas")
```

W celu graficznego przedstawienia zmian ekspresji 19 kluczowych genów wytypowanych w analizie interakcji, wygenerowano mapę ciepła (ang. heatmap) przy użyciu pakietu pheatmap. Proces objął następujące kroki konfiguracyjne:

- Normalizacja i skalowanie: Dane wejściowe (logCPM) poddano skalowaniu względem wierszy (scale = "row"), pozwoliło to na przedstawienie wyników jako Z-score. Dzięki czemu możliwe jest porównanie dynamiki genów o różnych bazowych poziomach ekspresji.
- Struktura wykresu: Wyłączono klastrowanie kolumn (cluster_cols = FALSE), aby zachować chronologiczny i logiczny układ próbek. W celu zwiększenia czytelności zastosowano przerwy wizualne (gaps_col) po każdej z grup eksperymentalnych (CTR 12h, EV 12h, CTR 24h).
- Adnotacje i kolorystyka: Do wykresu dołączono paski adnotacji informujące o przynależności próbek do grup oraz punktach czasowych. Zdefiniowano dedykowaną paletę barw: odcienie zieleni i turkusu dla grup oraz błękitu dla czasu. Skala ekspresji została przedstawiona w gradiencie od koloru niebieskiego (niska ekspresja) przez biały (średnia) do czerwonego (wysoka).
- Eksport wyników: Gotowy wykres został wyświetlony w środowisku R, a następnie wyeksportowany do pliku graficznego w formacie PNG o rozdzielczości 800x600 pikseli.

```
Analiza_3_heatmapa <- pheatmap(logCPM_sorted_analiza_3,
  scale = "row",
  cluster_cols = FALSE,
  cluster_rows = TRUE,
  gaps_col = c(4, 8, 12), #Przerwy po grupach: CTR_12h, EV_12h, CTR_24h
  annotation_col = annotation_col,
  annotation_names_col = FALSE,
  show_rownames = FALSE,
  show_colnames = TRUE,
  annotation_colors = list(
    Grupa = c(CTR = "chartreuse", EV = "darkturquoise"),
    Czas = c(`12` = "lightblue",`24` = "cadetblue")),
  color = colorRampPalette(c("blue", "white", "red"))(100),
  main = "Ekspresja genów - analiza dwuczynnikowa - treatment x time",
```

```
  name = "Z-score")

Analiza_3_heatmapa
dev.copy(png, "Analiza_3_heatmapa_dwuczynnikowa.png", width=800, height=600)
dev.off()
```

Porównanie wyników i analiza części wspólnych (Diagram Venna)

Fragment kodu służący do zestawienia wyników z modelu wieloczynnikowego, pozwala na identyfikację genów o unikalnych lub wspólnych profilach ekspresji.

1. W pierwszej kolejności z modelu wyciągane są listy genów, które przeszły rygorystyczny test statystyczny ($p < 0.01$)
    a. Efekt Główny Grupy (A): Geny różniące się między EV a CTR, niezależnie od czasu.
    b. Efekt Główny Czasu (B): Geny zmieniające się między 12h a 24h trwania hodowli.
    c. Efekt Interakcji (C): Geny, których reakcja na czynnik EV zmienia się w zależności od czasu (najważniejszy wynik analizy 3).
2. Przygotowanie danych do wizualizacji
    a. Tworzenie listy zbiorów: Wszystkie trzy listy nazw genów zostają połączone w jeden obiekt typu list, co jest wymagane do wygenerowania wykresu.
    b. Weryfikacja liczebności: Program wyświetla informację, ile genów znajduje się w każdym zbiorze, co pozwala na wstępną kontrolę danych przed rysowaniem.
3. Generowanie Diagramu Venna

Do stworzenia grafiki wykorzystano funkcję venn.diagram, która precyzyjnie pokazuje relacje między grupami:

    a. Wygląd: Zastosowano pastelową kolorystykę (Pastel1) oraz półprzezroczystość, aby nakładające się obszary były czytelne.
    b. Etykiety: Opisy kategorii zostały odpowiednio przesunięte i sformatowane, aby nie zasłaniały liczb wewnątrz kół.
    c. Zapis: Skrypt automatycznie usuwa stary plik (jeśli istniał) i zapisuje nową grafikę jako plik PNG.

```
#A.Efekt Główny Grupy (EV vs CTR)
top_treatment <- topTags(test_treatment_analiza_3, n = Inf, adjust.method =
"BH", p.value = 0.01)
genes_treatment <- rownames(top_treatment$table)

#B.Efekt Główny Czasu (24h vs 12h)
```

```
top_time <- topTags(test_time_analiza_3, n = Inf, adjust.method = "BH",
p.value = 0.01)
genes_time <- rownames(top_time$table)

#C.Efekt Interakcji (Już mamy z poprzedniej analizy)
genes_interaction <- rownames(top_interaction_analiza_3$table)

#Tworzenie listy zbiorów
lista_dwuczynnikowa <- list(
  "A: Efekt Grupy (EV vs CTR)" = genes_treatment,
  "B: Efekt Czasu (24h vs 12h)" = genes_time,
  "C: Efekt Interakcji (Grupa x Czas)" = genes_interaction)

#Weryfikacja liczebności
print(lapply(lista_dwuczynnikowa, length))
filename_dwuczynnikowy <- "Analiza_3_Diagram_Venna_Dwuczynnikowy.png"

#Usunięcie poprzedniego pliku
if (file.exists(filename_dwuczynnikowy)) file.remove(filename_dwuczynnikowy)

#Generowanie diagramu
venn.plot.dwuczynnikowy <- venn.diagram(
  x = lista_dwuczynnikowa,
  filename = filename_dwuczynnikowy,
  category.names = names(lista_dwuczynnikowa),
  main = "Analiza Dwuczynnikowa: Geny Efektów Głównych vs Interakcji",
  lwd = 2,
  fill = brewer.pal(3, "Pastel1"),
  alpha = 0.50,
  label.col = "black",
  cex = 1.0,
  cat.cex = 0.8,
  cat.fontfamily = "sans",
  #Dostosowanie pozycji etykiet
  cat.pos = c(-20, 20, 180),
  cat.dist = c(0.06, 0.06, 0.06))
```

Analiza 4: Dynamika zmian ekspresji genów w czasie

W czwartym etapie analizy skupiono się na zbadaniu, jak profil genetyczny komórek zmieniał się w trakcie trwania hodowli (pomiędzy 12. a 24. godziną) pod wpływem badanego czynnika. Kluczowe założenia analizy:

- Badanie dynamiki (Interakcja) - zamiast porównywać grupy tylko w jednym punkcie czasu, zastosowano model statystyczny badający interakcję (condition * time). Pozwoliło to wyłonić te geny, których reakcja na czynnik EV nie była stała, lecz zmieniała się wraz z upływem czasu hodowli.
- Wybór genów reagujących na czas - do dalszych badań zakwalifikowano geny, które wykazały najsilniejsze zmiany w odpowiedzi na leczenie właśnie w kontekście upływającego czasu (przy rygorystycznym progu istotności $p < 0.01$).

- Logiczny układ wyników - aby pokazać proces zmian w trakcie hodowli, dane na wykresach (heatmapach) zostały uporządkowane chronologicznie. Dzięki temu można zaobserwować, czy dany gen aktywował się już na początku (12h), czy jego główna reakcja nastąpiła dopiero w późniejszym etapie (24h).
- Analiza ta pozwoliła na wskazanie genów, które są odpowiedzialne za dynamiczną odpowiedź komórek na czynnik EV. Dzięki niej wiemy nie tylko „co" się zmieniło, ale również „kiedy" ta zmiana była najbardziej widoczna w trakcie trwania hodowli.

Poniższy fragment skryptu realizuje pełny proces statystycznej identyfikacji genów różnicujących warunki EV i CTR. W pierwszej kolejności dane są opakowywane w strukturę DGEList, gdzie następuje filtrowanie genów o zerowej ekspresji oraz normalizacja TMM mająca na celu zniwelowanie różnic w wielkości. Następnie program definiuje macierz opartą na zmiennej treatment, szacuje dyspersję biologiczną metodą robust i dopasowuje model uogólniony (GLM). Na podstawie dopasowanego modelu wykonywany jest test statystyczny, z którego wyodrębniane są geny o istotności $p < 0.01$ po korekcie FDR. Kod uzupełnia tabelę wyników o kolumnę klasyfikującą kierunek zmian (zwiększona ekspresja w EV lub CTR), a następnie generuje wykres Volcano przy użyciu biblioteki ggplot2 (Rysunek 11), wizualizując relację między siłą zmiany (logFC) a istotnością statystyczną.

```
dge_analiza_4 <- DGEList(counts = count_table, remove.zeros = TRUE)
dge_analiza_4 <- calcNormFactors(dge_analiza_4)

model_analiza_4 <- model.matrix(~ treatment, data = pheno_data)
dge_analiza_4 <- estimateDisp(dge_analiza_4, model_analiza_4, robust = TRUE)
fit_analiza_4 <- glmQLFit(dge_analiza_4, model_analiza_4)

test_EV_vs_CTR <- glmQLFTest(fit_analiza_4, coef = 2)
top_EV_vs_CTR <- topTags(test_EV_vs_CTR, n = Inf, adjust.method = 'BH',
p.value = 0.01)
genes_EV_vs_CTR <- rownames(top_EV_vs_CTR$table)

res_EV_vs_CTR <- test_EV_vs_CTR$table
res_EV_vs_CTR$FDR <- p.adjust(res_EV_vs_CTR$PValue, method="BH")
res_EV_vs_CTR$change <- "Bez znaczącej zmiany"
res_EV_vs_CTR$change <- ifelse(res_EV_vs_CTR$logFC > 1 & res_EV_vs_CTR$FDR <
0.05, "Ekspresja zwiększona w EV", res_EV_vs_CTR$change)
res_EV_vs_CTR$change <- ifelse(res_EV_vs_CTR$logFC < -1 & res_EV_vs_CTR$FDR <
0.05, "Ekspresja zwiększona w CTR", res_EV_vs_CTR$change)


Analiza_4_plot <- ggplot(res_EV_vs_CTR, aes(x=logFC, y=-log10(FDR),
color=change)) +
  geom_point(alpha=0.6, size=2) +
  scale_color_manual(values=c("Ekspresja zwiększona w EV"="red", "Ekspresja
zwiększona w CTR"="blue", "Bez znaczącej zmiany"="grey50")) +
```

```
  geom_vline(xintercept=c(-1,1), linetype="dashed", alpha=0.5) +
  geom_hline(yintercept=-log10(0.05), linetype="dashed", alpha=0.5) +
  xlab("log2 Fold Change") + ylab("-log10(FDR)") +
  ggtitle("EV vs CTR - Zmiany w ekspresji genów (wszystkie godziny)") +
  theme_minimal() + wyglad_volcano

Analiza_4_plot
dev.copy(png, "Analiza_4_volcano_EV_CTR.png", width=700, height=400)
dev.off()
```

Przedstawienie dynamiki zmian ekspresji genów różnicujących warunki EV i CTR na przestrzeni całego eksperymentu, wygenerowano zbiorczą mapę ciepła. Przygotowanie wizualizacji obejmuje następujące kroki:

- Ekstrakcja i filtrowanie danych - obliczono wartości logCPM dla zestawu danych, a następnie wyodrębniono z nich wyłącznie geny zidentyfikowane jako istotne statystycznie w teście globalnym.
- Organizowanie wykresu - próbki zostały posortowane według przynależności do grup (CTR, EV) oraz punktów czasowych (0h, 12h, 24h). Zastosowano przerwy wizualne (gaps_col), aby oddzielić od siebie poszczególne warianty badawcze.
- Normalizacja - dane poddano skalowaniu względem wierszy (Z-score), co umożliwiło wspólną prezentację genów o skrajnie różnych poziomach bazowej ekspresji w gradiencie kolorystycznym od niebieskiego (niska ekspresja) do czerwonego (wysoka ekspresja).
- Adnotacje i klastrowanie - dołączono panele informacyjne opisujące grupy i czas, stosując kolorystykę. Wyłączono klastrowanie kolumn w celu zachowania porządku czasu, pozostawiając klastrowanie wierszy, które grupuje geny o zbliżonych trendach odpowiedzi na badany czynnik.

```
   #heatmapa dla analizy 4#
#Obliczenie pełnej macierzy logCPM (na podstawie dge_analiza_4)
full_logCPM_analiza_4 <- cpm(dge_analiza_4, log = TRUE)

#Wyfiltrowanie logCPM dla istotnych genów
#genes_EV_vs_CTR to wynik testu glmQLFTest
logCPM_heatmap_analiza_4 <- full_logCPM_analiza_4[genes_EV_vs_CTR, ]

#Przygotowanie adnotacji kolumn (wykorzystanie pełnego pheno_data)
annotation_col_analiza_4 <- pheno_data[, c("condition", "time")]
colnames(annotation_col_analiza_4) <- c("Grupa", "Czas")
annotation_col_analiza_4$Czas <- as.factor(annotation_col_analiza_4$Czas)

sortowanie_analiza_4 <- order(annotation_col_analiza_4$Grupa,
  annotation_col_analiza_4$Czas)
```

```
sortowanie_analiza_4 <-
rownames(annotation_col_analiza_4)[sortowanie_analiza_4]

logCPM_sorted_analiza_4 <- logCPM_heatmap_analiza_4[, sortowanie_analiza_4]
annotation_col_sorted_analiza_4 <-
annotation_col_analiza_4[sortowanie_analiza_4, ]

Analiza_4_heatmapa <- pheatmap(logCPM_sorted_analiza_4,
  scale = "row",
  cluster_cols = FALSE,
  cluster_rows = TRUE,
  gaps_col = c(4, 8, 12, 16, 20), #Zakładając 4 repliki na punkt czasowy
  annotation_col = annotation_col_sorted_analiza_4,
  annotation_colors = list(
    Grupa = c(CTR = "chartreuse", EV = "darkturquoise"),
    Czas = c(`0` = "grey", `12` = "lightblue", `24` = "cadetblue")),
  annotation_names_col = FALSE,
  show_rownames = FALSE,
  show_colnames = TRUE,
  color = colorRampPalette(c("blue", "white", "red"))(100),
  main = "Heatmapa genów zmienionych (EV vs CTR - wszystkie godziny)",
  name = "Z-score")

Analiza_4_heatmapa
dev.copy(png, "Analiza_4_heatmapa_EV_vs_CTR_wszystkie_godziny.png", width=800,
height=1000)
dev.off()
```

Analiza 5 – Grupa kontrolna CTR

Wykonana analiza ma na celu zbadanie zmienności ekspresji genów wynikającej wyłącznie z upływu czasu trwania hodowli, bez wpływu czynnika badanego (EV).

Poniższy fragment kodu służy do zbadania zmian ekspresji genów zachodzących w czasie wyłącznie w grupie kontrolnej (CTR). Pozwala to na oddzielenie efektu upływu czasu od efektu traktowania pęcherzykami EV. Wyizolowano wyłącznie próbki CTR, usunięto geny o zerowej ekspresji, następnie przeprowadzono normalizacje w celu ujednolicenia.

Ustawiono grupę 0h jako punk odniesienia. W celu porównania późniejszych punktów czasowych.

Analiza edgeR – zbudowano model oparty na czynniku czasu i wykonano test QLF w celu wyłonienia genów istotnie zmieniających się. Równolegle przeprowadzono weryfikację metodą DESeq2 wykonując testy porównawcze dla tych samych punktów czasowych.

```
#Filtrowanie próbek CTR
pheno_data_ctr <- pheno_data[pheno_data$condition == "CTR", ]
pheno_data_ctr <- droplevels(pheno_data_ctr)

count_table_ctr <- count_table[, rownames(pheno_data_ctr)]
count_table_ctr <- as.matrix(count_table_ctr)
```

```
mode(count_table_ctr) <- "numeric"

dge_ctr <- DGEList(counts = count_table_ctr, remove.zeros = TRUE)
dge_ctr <- calcNormFactors(dge_ctr)

pheno_data_ctr$time <- as.factor(pheno_data_ctr$time)
pheno_data_ctr$time <- relevel(pheno_data_ctr$time, ref = "0")

#Model tylko na czas
design_ctr <- model.matrix(~ time, data = pheno_data_ctr)

dge_ctr <- estimateDisp(dge_ctr, design_ctr, robust = TRUE)
fit_ctr <- glmQLFit(dge_ctr, design_ctr)

#Test efektu czasu
test_time12_ctr <- glmQLFTest(fit_ctr, coef = 2) #Test 12h vs 0h
test_time24_ctr <- glmQLFTest(fit_ctr, coef = 3) #Test 24h vs 0h

top_time12_ctr <- topTags(test_time12_ctr, n = Inf, adjust.method = "BH",
p.value = 0.01)
top_time24_ctr <- topTags(test_time24_ctr, n = Inf, adjust.method = "BH",
p.value = 0.01)

pheno_ctr <- pheno_data_ctr
pheno_ctr$time <- as.factor(pheno_ctr$time)

dds_ctr <- DESeqDataSetFromMatrix(countData = count_table_ctr,
  colData = pheno_ctr,
  design = ~ time)
dds_ctr$time <- relevel(dds_ctr$time, ref = "0")
dds_ctr <- dds_ctr[rowSums(counts(dds_ctr)) > 1, ]
dds_ctr <- DESeq(dds_ctr)

res12 <- results(dds_ctr, contrast = c("time", "12", "0"))
res24 <- results(dds_ctr, contrast = c("time", "24", "0"))
```

W tej części kodu przygotowano wizualizację zmian ekspresji genów w grupie kontrolnej między 12 godziną a punktem zerowym hodowli. Utworzono klasyfikację genów na podstawie progów (|log2FC| >1 oraz FDR < 0.05). Przy użyciu biblioteki ggplot2 utworzono wykres Volcano, tak przygotowane dane wyeksportowano do pliku PNG.

```
#Volcano plot dla 12h vs 0h (edgeR)
res_time12 <- as.data.frame(test_time12_ctr$table)
res_time12$FDR <- p.adjust(res_time12$PValue, method="BH")
res_time12$change <- "Bez znaczącej zmiany"
res_time12$change <- ifelse(res_time12$logFC > 1 & res_time12$FDR < 0.05,
"Ekspresja zwiększona w 12h", res_time12$change)
res_time12$change <- ifelse(res_time12$logFC < -1 & res_time12$FDR < 0.05,
"Ekspresja zmiejszona w 12h", res_time12$change)

#Wykres dla analizy 5
ggplot(res_time12, aes(x=logFC, y=-log10(FDR), color=change)) +
  geom_point(alpha=0.6, size=2) +
  scale_color_manual(values=c("Ekspresja zwiększona w 12h"="red", "Ekspresja
zmiejszona w 12h"="blue", "Bez znaczącej zmiany"="grey50")) +
```

```
  geom_vline(xintercept=c(-1,1), linetype="dashed", alpha=0.5) +
  geom_hline(yintercept=-log10(0.05), linetype="dashed", alpha=0.5) +
  xlab("log2 Fold Change") + ylab("-log10(FDR)") +
  ggtitle("CTR: 12h vs 0h - Zmiany w ekspresji genów") +
  theme_minimal() + wyglad_volcano
dev.copy(png, "Analiza_5_volcano_CTR_12h_vs_0h.png", width=600, height=600)
dev.off()
```

Następnie dwukrotnie powtórzono czynności mające na celu wygenerowanie wizualizacji grup czasowych między:

- 24 godziną a punktem zerowych hodowli
- 24 a 12 godziną hodowli

```
#Volcano plot dla 24h vs 0h (edgeR)
res_time24 <- as.data.frame(test_time24_ctr$table)
res_time24$FDR <- p.adjust(res_time24$PValue, method="BH")
res_time24$change <- "Bez znaczącej zmiany"
res_time24$change <- ifelse(res_time24$logFC > 1 & res_time24$FDR < 0.05,
"Ekspresja zwiększona w 24h", res_time24$change)
res_time24$change <- ifelse(res_time24$logFC < -1 & res_time24$FDR < 0.05,
"Ekspresja zmiejszona w 24h", res_time24$change)

#24h vs 0h
ggplot(res_time24, aes(x=logFC, y=-log10(FDR), color=change)) +
  geom_point(alpha=0.6, size=2) +
  scale_color_manual(values=c("Ekspresja zwiększona w 24h"="red", "Ekspresja
zmiejszona w 24h"="blue", "Bez znaczącej zmiany"="grey50")) +
  geom_vline(xintercept=c(-1,1), linetype="dashed", alpha=0.5) +
  geom_hline(yintercept=-log10(0.05), linetype="dashed", alpha=0.5) +
  xlab("log2 Fold Change") + ylab("-log10(FDR)") +
  ggtitle("CTR: 24h vs 0h - Zmiany w ekspresji genów") +
  theme_minimal() + wyglad_volcano
dev.copy(png, "Analiza_5_volcano_CTR_24h_vs_0h.png", width=600, height=600)
dev.off()
```

```
#24h vs 12h
contrast_24vs12 <- c(0, -1, 1)
test_time24vs12_ctr <- glmQLFTest(fit_ctr, contrast = contrast_24vs12)
top_time24vs12_ctr <- topTags(test_time24vs12_ctr, n = Inf, adjust.method =
"BH", p.value = 0.01)
#Volcano plot dla 24h vs 12h (edgeR)
res_time24vs12 <- as.data.frame(test_time24vs12_ctr$table)
res_time24vs12$FDR <- p.adjust(res_time24vs12$PValue, method="BH")
res_time24vs12$change <- "Bez znaczącej zmiany"

#Oznaczenia zmian
#logFC > 1: Ekspresja zwiększona w 24h (względem 12h)
res_time24vs12$change <- ifelse(res_time24vs12$logFC > 1 & res_time24vs12$FDR
< 0.05, "Ekspresja zwiększona w 24h (vs 12h)", res_time24vs12$change)

#logFC < -1: Ekspresja zmniejszona w 24h (względem 12h)
res_time24vs12$change <- ifelse(res_time24vs12$logFC < -1 & res_time24vs12$FDR
< 0.05, "Ekspresja zmiejszona w 24h (vs 12h)", res_time24vs12$change)
```

```
ggplot(res_time24vs12, aes(x=logFC, y=-log10(FDR), color=change)) +
  geom_point(alpha=0.6, size=2) +
  scale_color_manual(values=c("Ekspresja zwiększona w 24h (vs 12h)"="red",
    "Ekspresja zmiejszona w 24h (vs 12h)"="blue",
    "Bez znaczącej zmiany"="grey50")) +
  geom_vline(xintercept=c(-1,1), linetype="dashed", alpha=0.5) +
  geom_hline(yintercept=-log10(0.05), linetype="dashed", alpha=0.5) +
  xlab("log2 Fold Change") + ylab("-log10(FDR)") +
  ggtitle("CTR: 24h vs 12h - Zmiany w ekspresji genów") +
  theme_minimal() + wyglad_volcano
dev.copy(png, "Analiza_5_volcano_CTR_24h_vs_12h.png", width=600, height=600)
dev.off()
```

Poniższy fragment służy do wizualizacji ekspresji genów zmieniających się w czasie w samej grupie kontrolnej (CTR). W celu utworzenia Heatmap obliczono wartości logCPM dla próbek kontrolnych oraz przefiltrowano po trzech porównaniach czasowych (12h vs 0h, 24h vs 0h, 24h vs 12h). Próbki ustawiono chronologicznie w celu ułatwienia obserwacji zmian. Wykorzystując funkcję „pheatmap" utworzono oddzielne wykresy ze skalowaniem względem wierszy (Z-Score). Kolor czerwony oznacza wzrost a kolor niebieski odpowiada za spadek aktywności genu. Każda heatmapa została zapisana jako oddzielny plik PNG.

```
#heatmapy dla analizy 5:
#obrobienie danych; Macierz logCPM dla wszystkich próbek CTR
logCPM_CTR <- cpm(dge_ctr, log = TRUE)

#Adnotacja kolumn (czas)
annotation_ctr <- pheno_data_ctr[, c("time"), drop = FALSE]
colnames(annotation_ctr) <- "Czas"
annotation_ctr$Czas <- as.factor(annotation_ctr$Czas)

#Kolory dla paska czasu
kolory_czas <- list(
  Czas = c(`0` = "grey",
    `12` = "lightblue",
    `24` = "darkblue")
)

#12h vs 0h
genes_12vs0 <- rownames(top_time12_ctr$table)
mat_12vs0 <- logCPM_CTR[genes_12vs0, ]

#Sortowanie próbek CTR (0h, 12h, 24h)
kolejnosc_analiza_5_heatmapy <- order(annotation_ctr$Czas)
kolejnosc_analiza_5_heatmapy <-
rownames(annotation_ctr)[kolejnosc_analiza_5_heatmapy]

mat_12vs0_sorted <- mat_12vs0[, kolejnosc_analiza_5_heatmapy]
annotation_12vs0_sorted <- annotation_ctr[kolejnosc_analiza_5_heatmapy, , drop
= FALSE]

pheatmap(mat_12vs0_sorted,
  scale = "row", cluster_cols = FALSE, cluster_rows = TRUE,
  gaps_col = c(4, 8),
```

```
  annotation_col = annotation_12vs0_sorted,
  annotation_colors = kolory_czas,
  annotation_names_col = FALSE,
  show_rownames = FALSE, show_colnames = TRUE,
  color = colorRampPalette(c("blue", "white", "red"))(100),
  name = "Z-Score",
  main = paste0("CTR: 12h vs 0h (Liczba genów: ", length(genes_12vs0), ")"))
dev.copy(png, "Analiza_5_Heatmap_CTR_12vs0.png", width=800, height=1000)
dev.off()

#24h vs 0h
genes_24vs0 <- rownames(top_time24_ctr$table)
mat_24vs0 <- logCPM_CTR[genes_24vs0, ]
mat_24vs0_sorted <- mat_24vs0[, kolejnosc_analiza_5_heatmapy] # Używamy tej
samej kolejności próbek
annotation_24vs0_sorted <- annotation_ctr[kolejnosc_analiza_5_heatmapy, , drop
= FALSE]

pheatmap(mat_24vs0_sorted,
  scale = "row", cluster_cols = FALSE, cluster_rows = TRUE,
  gaps_col = c(4, 8),
  annotation_col = annotation_24vs0_sorted,
  annotation_colors = kolory_czas,
  annotation_names_col = FALSE,
  show_rownames = FALSE, show_colnames = TRUE,
  color = colorRampPalette(c("blue", "white", "red"))(100),
  name = "Z-Score",
  main = paste0("CTR: 24h vs 0h (Liczba genów: ", length(genes_24vs0), ")"))
dev.copy(png, "Analiza_5_Heatmap_CTR_24vs0.png", width=800, height=1000)
dev.off()

#24h vs 12h
genes_24vs12 <- rownames(top_time24vs12_ctr$table)
mat_24vs12 <- logCPM_CTR[genes_24vs12, ]
mat_24vs12_sorted <- mat_24vs12[, kolejnosc_analiza_5_heatmapy] # Używamy tej
samej kolejności próbek
annotation_24vs12_sorted <- annotation_ctr[kolejnosc_analiza_5_heatmapy, ,
drop = FALSE]

pheatmap(mat_24vs12_sorted,
  scale = "row", cluster_cols = FALSE, cluster_rows = TRUE,
  gaps_col = c(4, 8),
  annotation_col = annotation_24vs12_sorted,
  annotation_colors = kolory_czas,
  annotation_names_col = FALSE,
  show_rownames = FALSE, show_colnames = TRUE,
  color = colorRampPalette(c("blue", "white", "red"))(100),
  name = "Z-Score",
  main = paste0("CTR: 24h vs 12h (Liczba genów: ", length(genes_24vs12), ")"))
dev.copy(png, "Analiza_5_Heatmap_CTR_24vs12.png", width=800, height=1000)
dev.off()
```

Ostatnią częścią analizy 5 było wygenerowanie diagramu Venna. Na podstawie wcześniej przygotowanych tabel wyodrębniono listy genów niezbędne do stworzenia zestawienia. Za pomocą funkcji „print” wyświetlono w konsoli liczbę genów dla

poszczególnych przedziałów czasowych. W celu zapisu wyniku zdefiniowano nazwę pliku, a przy użyciu instrukcji warunkowej „if” sprawdzano jego obecność w katalogu roboczym – w przypadku istnienia starszej wersji, była ona usuwana i zastępowana nowo wygenerowanym diagramem.

```
#diagram venna
#Wyciągnięcie list genów
lista_genow <- list(
  "12h vs 0h" = rownames(top_time12_ctr$table),
  "24h vs 0h" = rownames(top_time24_ctr$table),
  "24h vs 12h" = rownames(top_time24vs12_ctr$table))

#Sprawdzenie ilosci genów w danym zbiorze
print(lapply(lista_genow, length))

filename <- "Analiza_5_Diagram_Venna_CTR_Czas.png"
#Usunięcie poprzedniego pliku
if (file.exists(filename)) file.remove(filename)

#Generowanie diagramu
venn.plot <- venn.diagram(
  x = lista_genow,
  filename = filename,
  category.names = names(lista_genow),
  main = "Geny istotnie zmienione w czasie (Grupa CTR)",
  #Wygląd kół
  lwd = 2,
  fill = brewer.pal(3, "Pastel1"),
  alpha = 0.50,
  label.col = "black",
  cex = 1, #Wielkość tekstu
  fontfamily = "sans",
  #Wygląd kategorii
  cat.cex = 1, #Wielkość nazw kategorii
  cat.pos = c(-22, 20, 180),
  cat.dist = c(0.06, 0.06, 0.06))
```

Analiza trendu wyników modelu wieloczynnikowego

Przygotowanie danych oraz wygenerowanie wykresu trendu dla 19 genów wyłonionych w analizie 3. Z tabeli genes_interaction_analiza_3 pobrano wszystkie geny, przypisano je do nowej zmiennej. Za pomocą funkcji „melt” przekształcono tabelę do formatu długiego. Usunięto nadmiarowe znaki z ID genów oraz dołączono metadane o grupach i czasie za pomocą funkcji „merge”. Za pomocą biblioteki ggplot2 utworzono wykres punktowy z podziałem na panele (facet_wrap) dla każdego z genów.

```
interakcja_data <- df_log[genes_interaction_analiza_3[1:19], ]
interakcja_melted <- melt(interakcja_data)
colnames(interakcja_melted) <- c("Gene", "Sample", "LogCPM")
#Usunięcie nadmiarowych znaków
```

```
interakcja_melted$Gene <- gsub("\\|.*", "", interakcja_melted$Gene)
#złączenie tabel
interakcja_melted <- merge(interakcja_melted, pheno_data, by.x="Sample",
by.y="row.names")

#Wykres trendu czasowego dla kluczowych 19 genów
wykres_trendu_19_genow <- ggplot(interakcja_melted, aes(x=time, y=LogCPM,
color=treatment, group=treatment)) +
  geom_point() +
  facet_wrap(~Gene, scales="free_y") +
  theme_bw() +
  ggtitle("Trend ekspresji genów interakcji (Zależność EV od czasu)") +
  labs(,
    x = "Czas (h)",
    y = "Poziom ekspresji (LogCPM)")

wykres_trendu_19_genow
dev.copy(png, "Wykres_trendu_19_kluczowych_genow.png", width=700, height=500)
dev.off()
```

Selekcja genów o najwyższym współczynniku zróżnicowania ekspresji

W celu dokładniejszego zbadania dynamiki zmian, wybrano trzy geny o najsilniejszym zróżnicowaniu trendów. Zdefiniowano wektor (wybrane_geny) z konkretnymi identyfikatorami Ensembl ID, a następnie pobrano odpowiadające im pełne nazwy i wartości ekspresji z głównego zbioru danych. Dane zostały sformatowane i złączone z informacjami fenotypowymi pochodzącymi z tabelu pheno_data. Następnie utworzono wykres punktowy z dostosowaną kolorystyką (fiolet dla EV, zieleń dla CTR), który w czytelny sposób prezentuje różnice w profilach ekspresji wybranych genów pomiędzy badanymi warunkami w punktach czasowych 0h, 12h i 24h.

```
#Wybór genów
wybrane_geny <- c("ENSG00000103522", "ENSG00000133110", "ENSG00000196628")
pelne_nazwy_3 <- genes_interaction_analiza_3[grep(paste(wybrane_geny,
collapse="|"), genes_interaction_analiza_3)]

#Przygotowanie tabeli
interakcja_data_3 <- df_log[pelne_nazwy_3, ]
interakcja_melted_3 <- reshape2::melt(interakcja_data_3)
colnames(interakcja_melted_3) <- c("Gene", "Sample", "LogCPM")
interakcja_melted_3$Gene <- gsub("\\|.*", "", interakcja_melted_3$Gene)

#Łączenie z pheno_data
interakcja_melted_3 <- merge(interakcja_melted_3, pheno_data, by.x="Sample",
by.y="row.names")

interakcja_melted_3$condition <- as.factor(interakcja_melted_3$condition)
interakcja_melted_3$time <- as.factor(interakcja_melted_3$time)

#Generowanie wykresu
```

```
wykres_trendu_top_3_geny <- ggplot(interakcja_melted_3,
  aes(x = time, y = LogCPM, color = condition, group = condition)) +
  geom_point(size = 3, alpha = 0.8) +
  facet_wrap(~Gene, scales = "free_y") +
  theme_bw() +
  scale_color_manual(values = c("CTR" = "chartreuse3", "EV" = "darkorchid1"))
+
  labs(
    title = "Dynamika ekspresji wybranych genów",
    subtitle = "Porównanie trendów: EV vs CTR (Top 3)",
    x = "Czas (h)",
    y = "Poziom ekspresji (LogCPM)",
    color = "Grupa")

wykres_trendu_top_3_geny

dev.copy(png, "Wykres_trendu_top_3_geny.png", width=700, height=500)
dev.off()
```

Geny unikatowe dla analizy wieloczynnikowej

Ostatnia sekcja kodu ukierunkowana jest na wizualizację dwóch unikalnych genów, które w diagramie Venna nie wykazywały pokrycia z efektami głównymi grupy ani czasu. Na podstawie zdefiniowanych ID wyodrębniono dane dla dwóch specyficznych genów. Procedura przygotowania ramki danych (melt, czyszczenie nazw, merge) została powtórzona w celu zapewnienia kompatybilności z funkcjami graficznymi. Wygenerowano wizualizację trendów czasowych dla tej pary genów, co pozwala na ocenę ich specyficznej reakcji na czynnik EV, która odróżnia je od pozostałych istotnych statystycznie genów.

```
#Wybór genów
wybrane_unikalne <- c("ENSG00000108551", "ENSG00000196628")
wybrane_unikalne_pelne_nazwy_3 <-
genes_interaction_analiza_3[grep(paste(wybrane_unikalne, collapse="|"),
genes_interaction_analiza_3)]

#Przygotowanie tabeli
wybrane_unikalne_interakcja_data_3 <- df_log[wybrane_unikalne_pelne_nazwy_3, ]
wybrane_unikalne_interakcja_data_3 <-
reshape2::melt(wybrane_unikalne_interakcja_data_3)
colnames(wybrane_unikalne_interakcja_data_3) <- c("Gene", "Sample", "LogCPM")
wybrane_unikalne_interakcja_data_3$Gene <- gsub("\\|.*", "",
wybrane_unikalne_interakcja_data_3$Gene)

#Łączenie z pheno_data
wybrane_unikalne_interakcja_data_3 <-
merge(wybrane_unikalne_interakcja_data_3, pheno_data, by.x="Sample",
by.y="row.names")

wybrane_unikalne_interakcja_data_3$condition <-
as.factor(wybrane_unikalne_interakcja_data_3$condition)
wybrane_unikalne_interakcja_data_3$time <-
as.factor(wybrane_unikalne_interakcja_data_3$time)
```

```
#Generowanie wykresu
wykres_trendu_wybrane_unikalne <- ggplot(wybrane_unikalne_interakcja_data_3,
  aes(x = time, y = LogCPM, color = condition, group = condition)) +
  geom_point(size = 3, alpha = 0.8) +
  facet_wrap(~Gene, scales = "free_y") +
  theme_bw() +
  scale_color_manual(values = c("CTR" = "chartreuse3", "EV" = "darkorchid1"))
+
  labs(
    title = "Dynamika ekspresji wybranych genów",
    subtitle = "Porównanie trendów: EV vs CTR (Unikalne)",
    x = "Czas (h)",
    y = "Poziom ekspresji (LogCPM)",
    color = "Grupa")

wykres_trendu_wybrane_unikalne

dev.copy(png, "wykres_trendu_wybrane_unikalne.png", width=700, height=500)
dev.off()
```

Etap IV: Eksport wyników

Wyeksportowanie wyników do ostatecznej obróbki przebiegało według następujących instrukcji. Utworzono funkcję, która pozwala na usunięcie za pomocą polecenia gsub nadmiarowe znaki znajdujące się po znaku „|" w nazwie ID genów. Przygotowano odpowiednio złożony data frame tak aby znalazły się w nim oczyszczone z niechcianych znaków nazwy. Dane znajdujące się w „df_to_export" złączono z wybranymi kolumnami z tabeli „allgenes.Ensembl". W kolejnym kroku wywołano zapis wraz z informacja zwrotną na konsoli o utworzonym zapisie. Ostatnim krokiem było wywołanie funkcji na poszczególnych tabelach oraz nadanie im właściwych nazw.

```
#Pobieranie listy genów, dołączenie opisów i zapis do pliku
export_heatmap_genes <- function(gene_list, fileName) {
  clean_ids <- gsub("\\|.*", "", gene_list)

  df_to_export <- data.frame(ensembl_gene_id = clean_ids, stringsAsFactors =
FALSE)
  #Dołączenie adnotacji z tabeli allgenes.Ensembl
  df_final <- merge(df_to_export,
    allgenes.Ensembl[, c("ensembl_gene_id", "external_gene_name",
"gene_biotype", "description")],
    by = "ensembl_gene_id",
    all.x = TRUE)

  write.csv2(df_final, file = fileName, row.names = FALSE)
  cat("Zapisano:", fileName, "- liczba genów:", nrow(df_final), "\n")}

#Analiza 1 (12h)
export_heatmap_genes(sig_genes_names_12, "Geny_Heatmapa_A1_12h.csv")

#Analiza 2 (24h)
```

```
export_heatmap_genes(sig_genes_names_24, "Geny_Heatmapa_A2_24h.csv")

#Analiza 3 (Interakcja - 19 genów)
export_heatmap_genes(genes_interaction_analiza_3,
"Geny_Heatmapa_A3_Interakcja.csv")

#Analiza 4 (Globalna EV vs CTR)
export_heatmap_genes(genes_EV_vs_CTR, "Geny_Heatmapa_A4_Globalna.csv")

#Analiza 5 (CTR - 3 heatmapy)
export_heatmap_genes(genes_12vs0, "Geny_Heatmapa_A5_12vs0.csv")
export_heatmap_genes(genes_24vs0, "Geny_Heatmapa_A5_24vs0.csv")
export_heatmap_genes(genes_24vs12, "Geny_Heatmapa_A5_24vs12.csv")
```

Wyniki – interpretacja

Analiza PCA

Na Rysunku 1 przedstawiono rzut próbek na płaszczyznę dwóch pierwszych głównych składowych, które łącznie wyjaśniają 28,2% całkowitej wariancji zbioru danych.

- Pierwsza główna składowa (PC1): Wyjaśnia 16,3% zmienności i odpowiada głównie za separację próbek ze względu na warunek eksperymentalny (traktowanie). Próbki z grupy kontrolnej (CTR) są wyraźnie oddzielone wzdłuż osi X od próbek poddanych działaniu pęcherzyków zewnątrzkomórkowych (EV), co potwierdza istotny wpływ czynnika EV na globalny profil komórek.
- Druga główna składowa (PC2): Wyjaśnia 11,9% zmienności i odzwierciedla dynamikę czasu trwania eksperymentu. Można zauważyć wyraźny trend przesunięcia próbek w pionie wraz z upływem czasu (od 0h przez 12h do 24h).

Analiza wykresu pozwala stwierdzić wysoką jakość danych doświadczalnych. Powtórzenia biologiczne w ramach tych samych grup (np. CTR_0h czy EV_24h) wykazują tendencję do bliskiego sąsiedztwa w przestrzeni PCA, co świadczy o dobrej powtarzalności eksperymentu i niskim poziomie szumu technicznego. Wyraźna separacja klastrów CTR i EV sugeruje, że wprowadzenie pęcherzyków zewnątrzkomórkowych indukuje systematyczne i mierzalne zmiany w ekspresji genów.

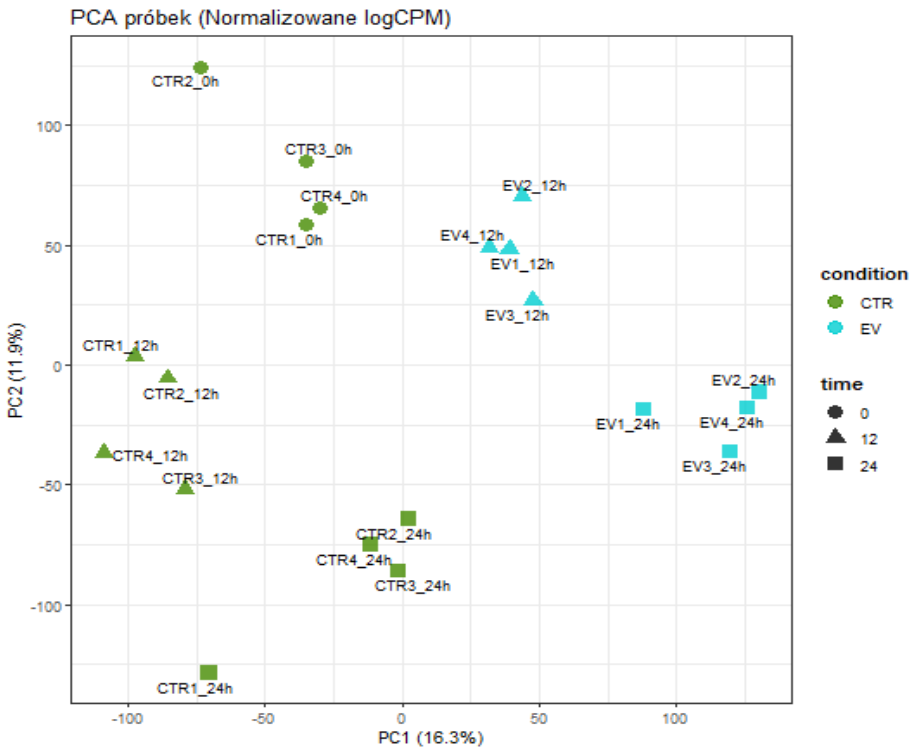

Figure 1 PCA analysis

Rysunek 1 Analiza PCA

Wykres czasu nMDS

Interpretacja wyników nMDS

Na wygenerowanym wykresie nMDS (Rysunek 2) zastosowano elipsy ufności, które pozwalają na wizualne potwierdzenie statystycznej odrębności grup CTR i EV.

Separacja grup: Podobnie jak w analizie PCA, próbki kontrolne i badawcze tworzą wyraźnie oddzielone od siebie skupiska, co potwierdza, że czynnik traktowania pęcherzykami EV jest dominującym źródłem zmienności w badanym układzie.

Kluczowym wskaźnikiem wiarygodności tej metody jest wartość stresu (stress value), która dla przeprowadzonej analizy wynosi 0,101.

```
> print(paste("nMDS Stress Value:", nmds_results$stress))
[1] "nMDS Stress Value: 0.101385985068651"
```

W literaturze przyjmuje się, że wynik poniżej 0,2 jest akceptowalny, natomiast wartość bliska 0,1 świadczy o dobrym dopasowaniu [12] i oznacza, że dwuwymiarowy wykres rzetelnie odzwierciedla rzeczywiste, wielowymiarowe relacje między próbkami.

Figure 2 nMDS analysis

Rysunek 2 Analiza nMDS

Wizualizacja pozwala na wyciągnięcie kluczowych wniosków dotyczących dynamiki badanego procesu:

- Wyraźna separacja punktów czasowych: elipsy reprezentujące próbki z czasu 0h (kolor czerwony) oraz 24h (kolor niebieski) są niemal całkowicie rozdzielone w przestrzeni dwuwymiarowej. Świadczy to o tym, że profil ekspresji genów ulega systematycznej i głębokiej przebudowie wraz z czasem trwania eksperymentu.
- Charakter przejściowy punktu 12h: elipsa dla czasu 12h (kolor zielony) zajmuje pozycję pośrednią i częściowo nakłada się na oba skrajne punkty czasowe. Taka struktura wykresu jest biologicznie uzasadniona – odzwierciedla ona progresywny charakter zmian.
- Wpływ czasu jako głównego źródła zmienności: Kierunek przesunięcia klastrów wzdłuż osi NMDS1 i NMDS2 sugeruje, że czas jest jednym z dominujących czynników kształtujących różnice w danych. Potwierdza to zasadność przeprowadzenia zaawansowanej analizy wieloczynnikowej (Analiza 3), która pozwoli na oddzielenie efektów czysto czasowych od zmian wywołanych traktowaniem EV.

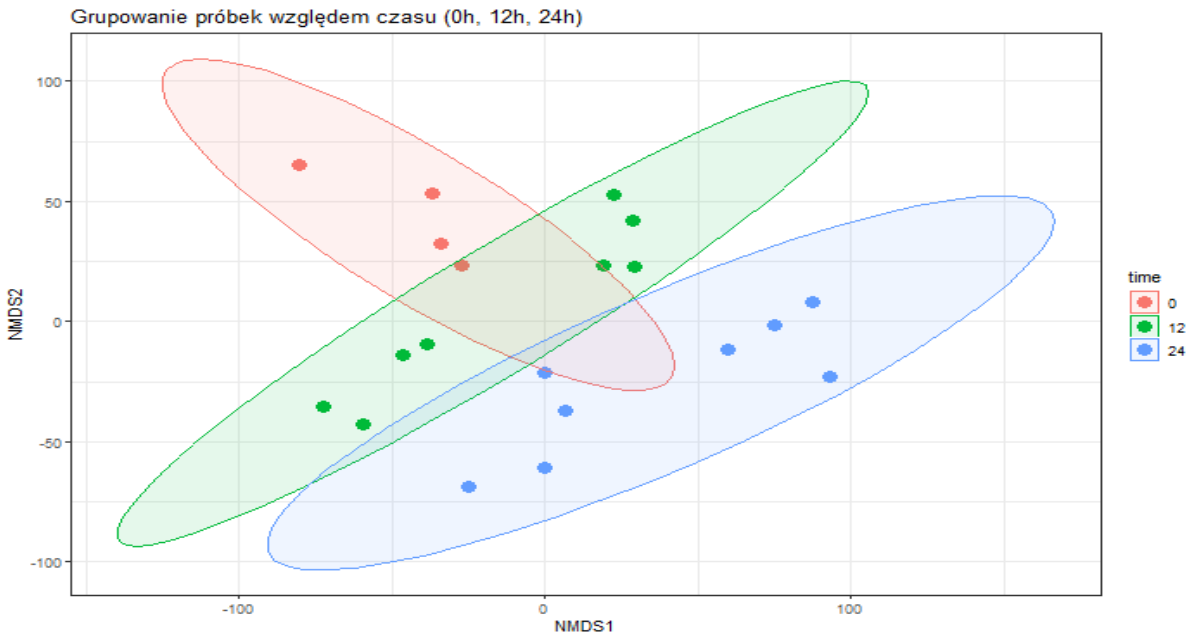

Figure 3 nMDS analysis - time factor

Rysunek 3 nMDS czas

Klastrowanie Hierarchiczne

Zastosowana metoda Warda (wariant ward.D2), stanowi algorytm aglomeracyjny oparty na kryterium minimalnej wariancji. Wybór tej metody był podyktowany jej wysoką skutecznością w identyfikacji zwartych i wyraźnie odseparowanych klastrów. Algorytm na każdym kroku łączy parę skupień, która minimalizuje wzrost całkowitej sumy kwadratów odchyleń wewnątrz grup, co pozwala na rzetelne odzwierciedlenie podobieństwa biologicznego między replikatami w ramach warunków EV i CTR.

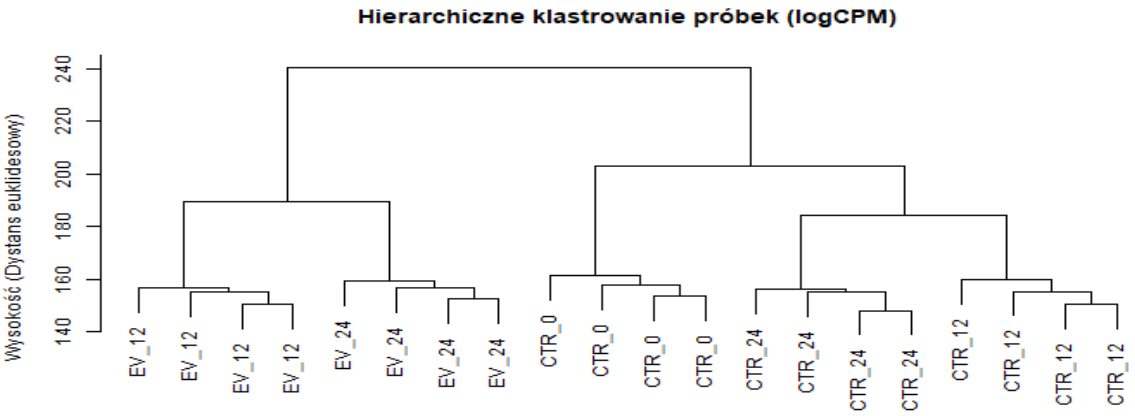

Figure 4 Hierarchical clustering

Rysunek 4 Klastrowanie hierarchiczne

Wyraźna separacja klastrów CTR i EV oraz brak próbek odstających potwierdzają wysoką jakość biologiczną materiału oraz prawidłowość procedury laboratoryjnej. Potwierdzona w tym kroku silna struktura danych stanowi podstawę do przeprowadzenia testów różnicowej ekspresji genów (DGE) w kolejnych krokach analizy.

Analiza 1 - Porównanie warunków EV kontra CTR w 12 godzinie trwania hodowli

Analiza 1 ma na celu umożliwienie szybkiego spojrzenia na różnorodność próbek i ich reakcję. Do analizy 1 wykorzystano próbki z hodowli po 12 godzinach jej trwania. Wykres Volcano (Rysunek 5) pozwala na jednoczesną ocenę statystycznej istotności zmian oraz ich siły biologicznej:

- Oś X (log2 Fold Change) ukazuje kierunek i siłę zmiany aktywności genów. Wartości dodatnie oznaczają wzrost ekspresji w grupie EV, natomiast ujemne jej spadek względem grupy CTR.
- Oś Y (-log10 padj) odzwierciedla pewność statystyczną; im wyżej znajduje się punkt, tym mniejsze jest prawdopodobieństwo, że zaobserwowana zmiana wynika z przypadku. Geny uznane za istotne statystycznie (progi $p < 0.05$ oraz bezwzględną zmianę krotności log2FC > 1) zostały wyróżnione kolorami:
    - Kolor czerwony - geny o zwiększonej ekspresji w grupie EV.
    - Kolor niebieski - geny o zmniejszonej ekspresji w grupie EV.
    - Kolor szary - geny, które nie spełniły założonych kryteriów istotności lub siły zmiany.

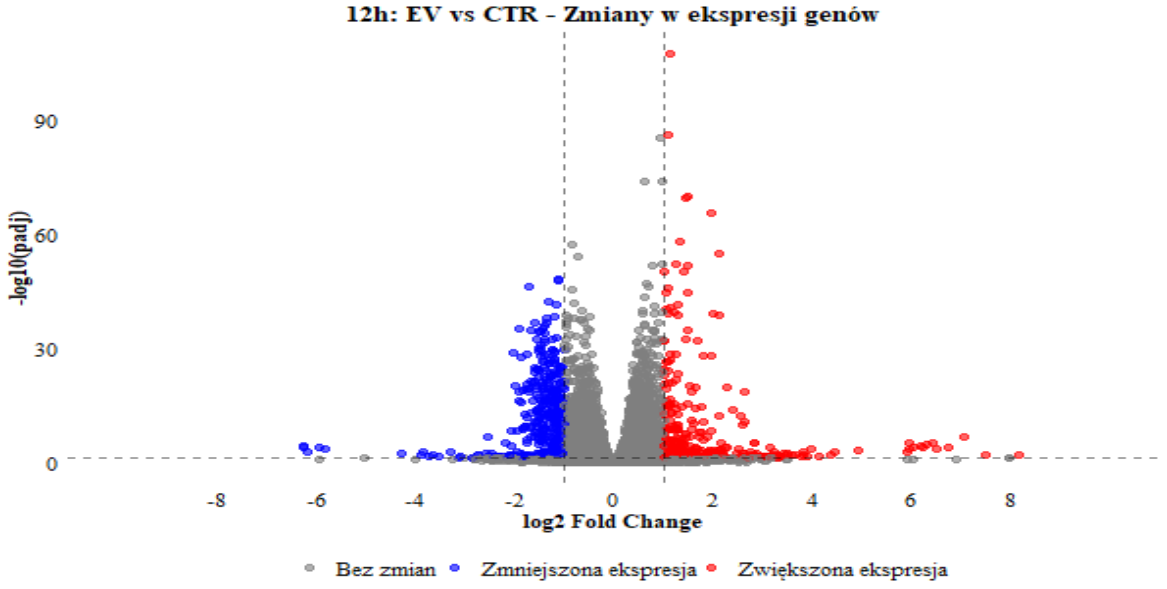

Figure 5 Analysis 1 Volcano Plot

Rysunek 5 Analiza 1 Volcano Plot

Heatmapa (Rysunek 6) stanowi graficzną prezentację poziomów ekspresji genów wytypowanych jako istotne w 12 godzinie eksperymentu. Pozwala na ocenę powtarzalności wyników wewnątrz grup oraz identyfikację kierunków zmian.

- Skala kolorystyczna (Z-score): Intensywność kolorów odzwierciedla względny poziom aktywności genów. Kolor czerwony oznacza wysoką ekspresję, natomiast kolor niebieski oznacza niską ekspresję w danej próbce.
- Grupowanie próbek (kolumny): Na górze wykresu widoczny jest wyraźny podział na grupę kontrolną (CTR – kolor zielony) oraz grupę traktowaną EV (kolor turkusowy). Spójność kolorów w pionowych blokach świadczy o wysokiej powtarzalności biologicznej między replikatami.
- Klastrowanie genów (wiersze): Geny o podobnych profilach reakcji są automatycznie grupowane obok siebie (drzewo klastrowania po lewej stronie). Wyraźnie widoczne są dwa główne bloki: geny, które pęcherzyki EV silnie aktywują (czerwone w grupie EV), oraz te, które ulegają wyciszeniu (niebieskie w grupie EV).

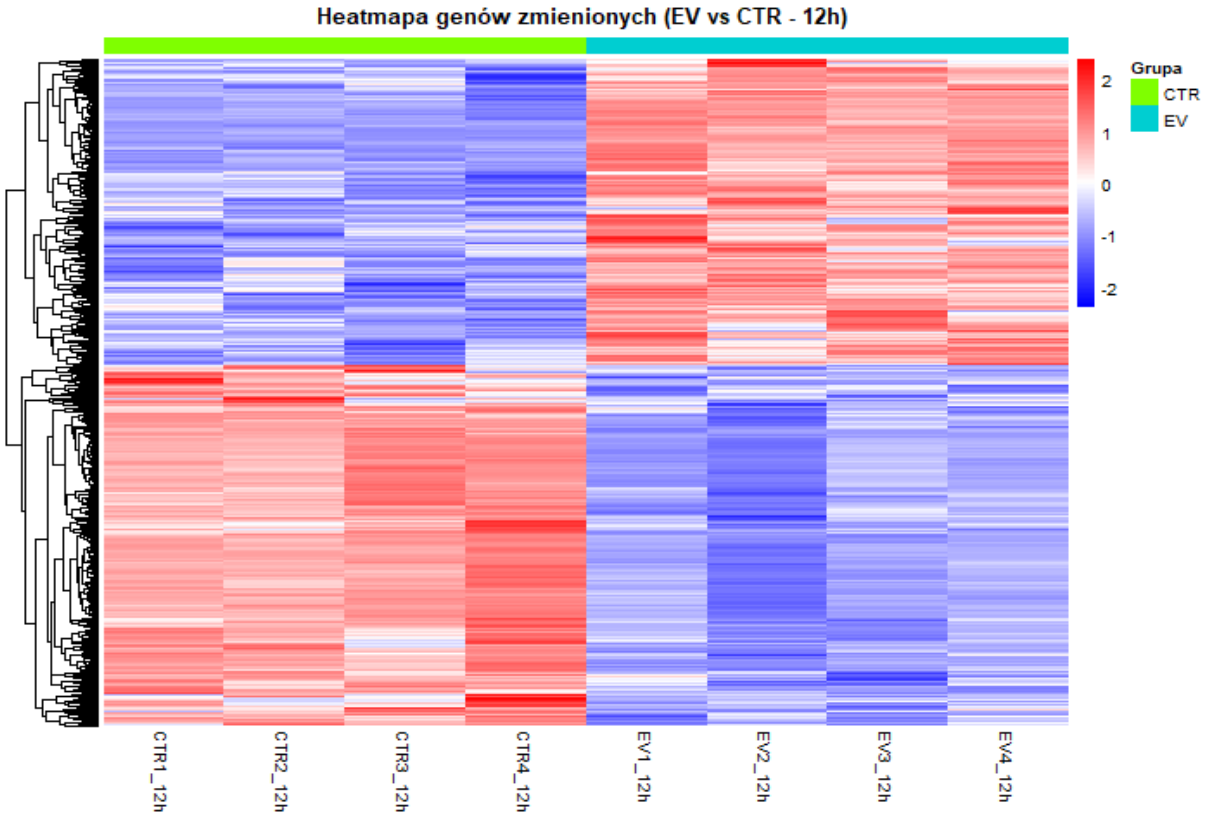

Figure 6 Analysis 1 Heatmap

Rysunek 6 Analiza 1 Heatmap

Analiza 2 Porównanie warunków EV kontra CTR w 24 godzinie trwania hodowli

Analiza 2 jest analogicznym do analizy 1 porównaniem próbek. Różnicą natomiast jest zastosowany czas hodowli. W analizie 2 zweryfikowano próbki z hodowli po 24 godzinach jej trwania. Wykres Volcano (Rysunek 7) przedstawia istotność i siłę zmian ekspresji genów po 24 godzinach eksperymentu.

- Osie wykresu - oś X (log2FC) reprezentuje kierunek zmian (dodatnie – wzrost w EV, ujemne – spadek w EV), a oś Y (-log10 padj) stopień wiarygodności statystycznej.
- Kolorystyka: Zastosowano te same kryteria selekcji co w Analizie 1 (log2FC > 1, p < 0.05). Punkty czerwone i niebieskie wskazują geny o najsilniejszej, potwierdzonej statystycznie reakcji na traktowanie pęcherzykami EV w 24. godzinie.

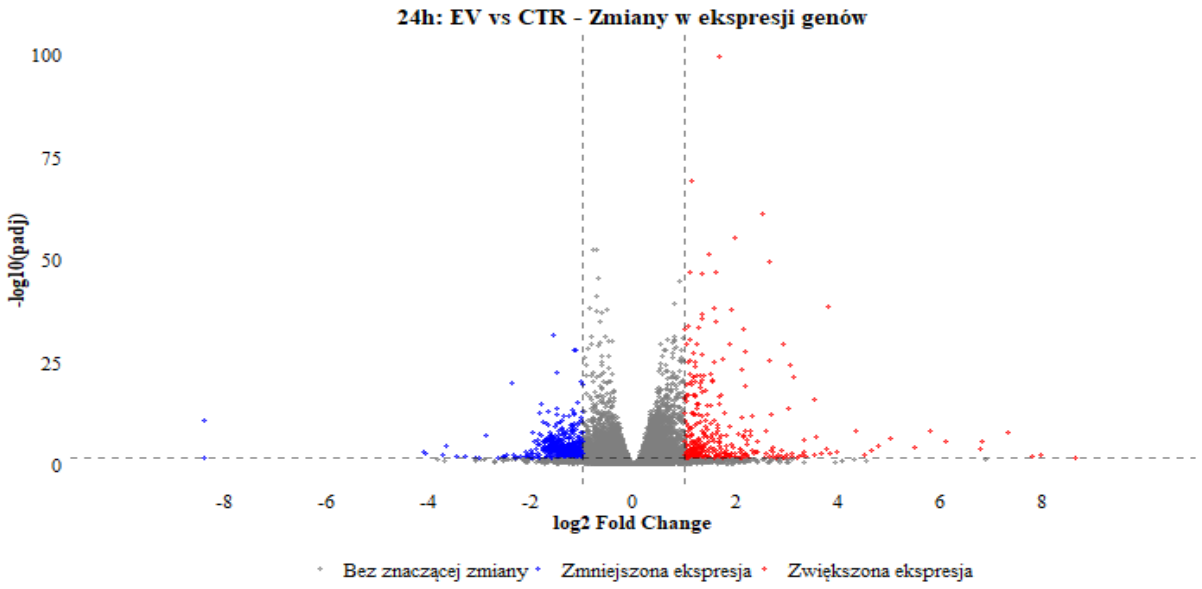

Figure 7 Analysis 2 Volcano Plot

Rysunek 7 Analiza 2 Volcano Plot

Heatmapa (Rysunek 8) obrazuje profile ekspresji genów różnicujących grupy po 24 godzinach.

- Skala Z-score: Kolor czerwony odpowiada wysokiej, a niebieski niskiej aktywności genów, co pozwala na wizualne porównanie dynamiki zmian między próbkami.
- Spójność grupowa: Podział kolumn na bloki CTR (zielony) i EV (turkusowy) potwierdza utrzymującą się po 24 godzinach powtarzalność między replikatami biologicznymi.

- Klastrowanie: Układ wierszy grupuje geny o zbliżonym wzorcu odpowiedzi, uwidaczniając stabilne bloki genów aktywowanych oraz wyciszanych przez czynnik EV w późnej fazie eksperymentu.

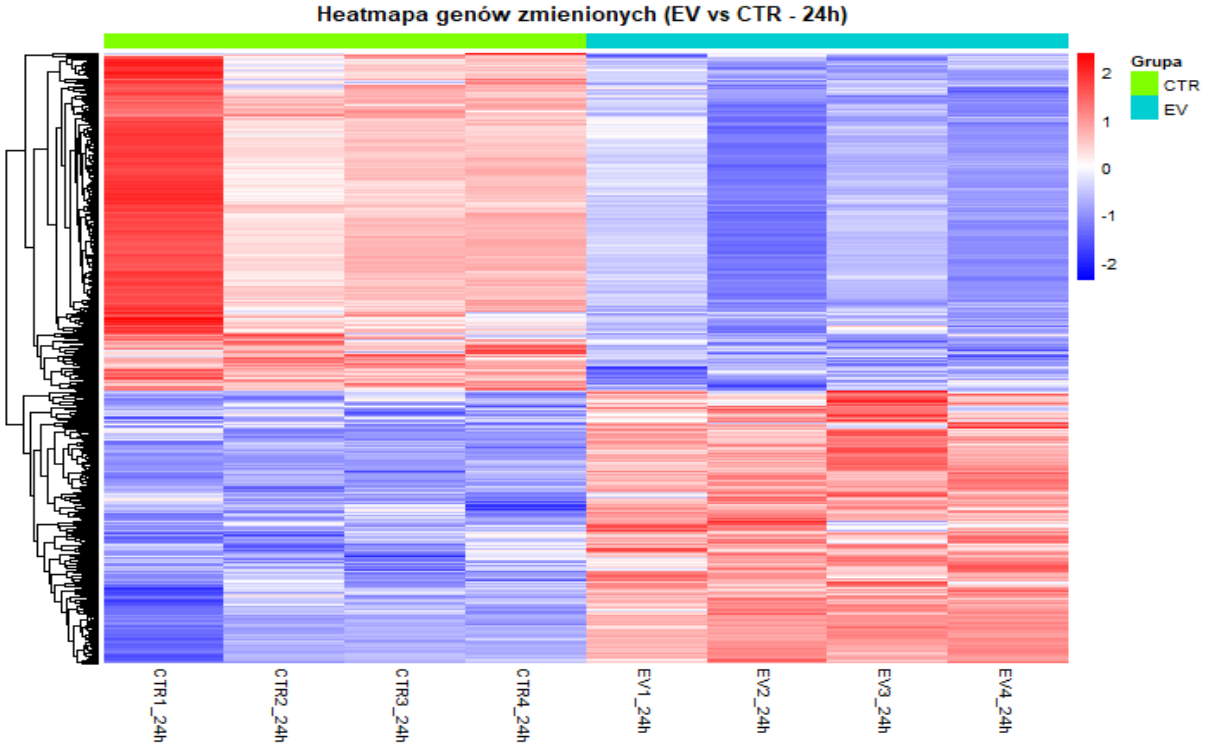

Figure 8 Analysis 2 Heatmap

Rysunek 8 Analiza 2 Heatmap

Analiza 3 Model wieloczynnikowy i analiza interakcji

Model wieloczynnikowy dynamiki zmian ekspresji 19 genów wykazujących istotną statystycznie interakcję (treatment * time), wygenerowano mapę ciepła (Rysunek 9). Heatmapa (Rysunek 9) ujawnia złożone wzorce ekspresji, które nie byłyby widoczne w prostych porównaniach parowych. Geny zostały pogrupowane w bloki o zbliżonym profilu reakcji, dzięki czemu widoczne są grupy genów, których aktywność wzrasta w grupie EV dopiero po 24 godzinach, podczas gdy w 12. godzinie pozostaje na poziomie zbliżonym do kontroli (lub odwrotnie). Wizualizacja pozwala na potwierdzenie, że odpowiedź na badany czynnik nie jest stała, lecz ewoluuje w czasie. Zmiana kolorów (od niebieskiego do czerwonego) wyraźnie wskazuje, że dla wyselekcjonowanych genów wpływ pęcherzyków zewnątrzkomórkowych jest uzależniony od czasu trwania hodowli. Odcienie koloru czerwonego pozwalają na zweryfikowanie jak silna była aktywność danego genu w okresie czasowym. Odcienie koloru niebieskiego pozwalają określić jak aktywność genu spadała w czasie.

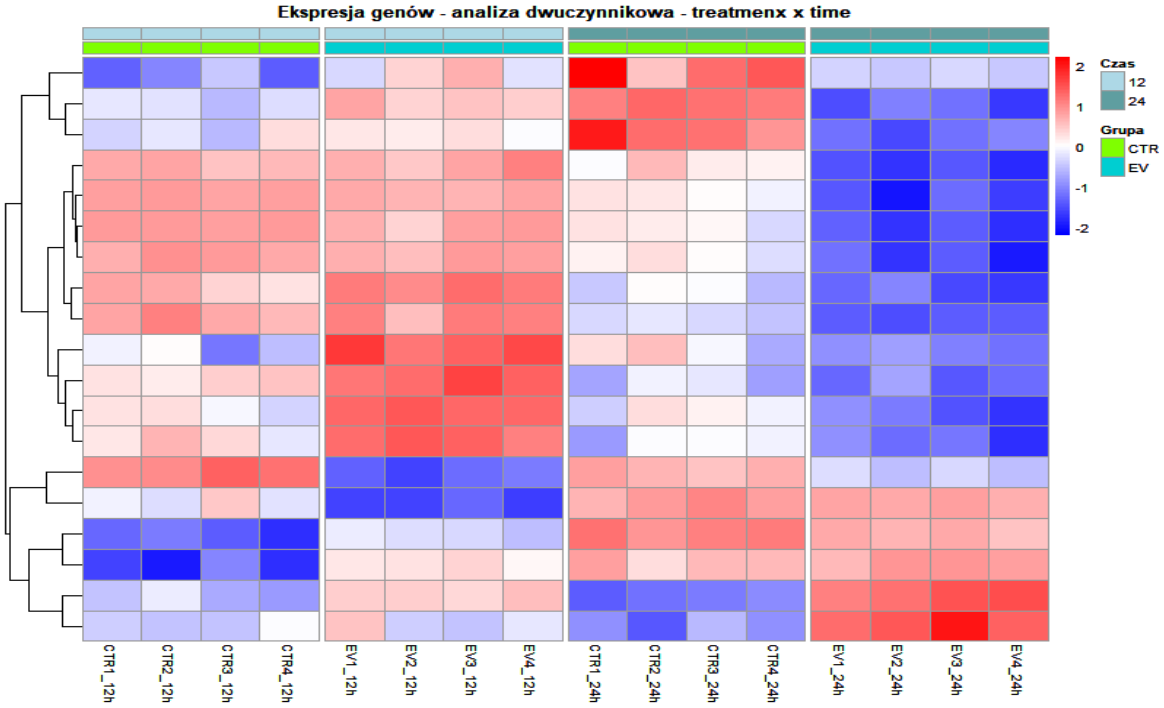

Figure 9 Analysis 3 Heatmap

Rysunek 9 Analiza 3 Heatmap

Utworzenie diagramu Venna w celu zweryfikowania pokrycia efektu interakcji względem efektu grupy oraz czasu. Utworzony diagram ujawnia dwie kluczowe informacje:

- Istnieje 6 genów, których udział zauważono zarówno w efekcie grupy jak i czasu
- Istnieją dwa geny, które swojego udziału nie wiążą z żadną z grup

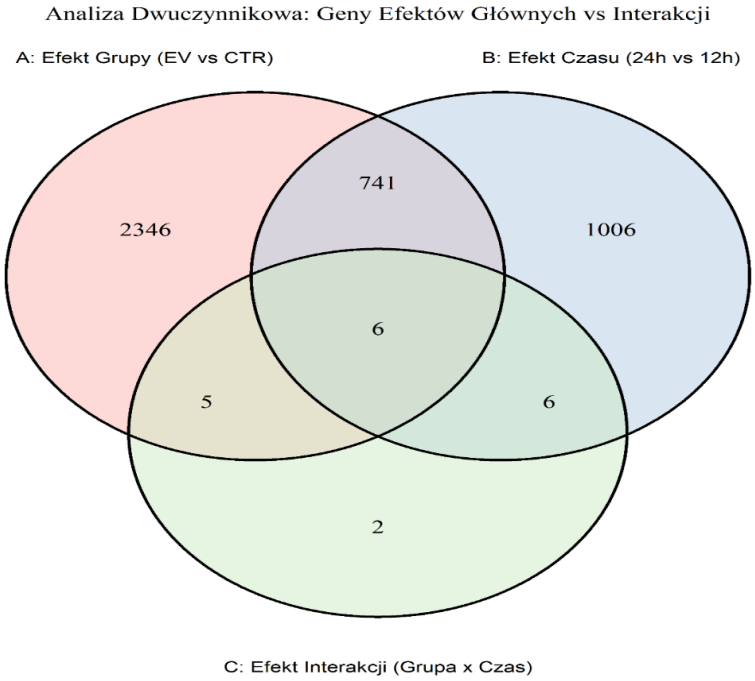

Figure 10 Analysis 3 Venn Diagram

Rysunek 10 Analiza 3 Diagram Venna

Analiza 4: Dynamika zmian ekspresji genów w czasie

Głównym aspektem tej analizy było uchwycenie profilu transkryptomicznego w kontekście upływu czasu. Wyniki zostały zwizualizowane za pomocą dwóch metod graficznych.

Wykres Volcano (Rysunek 11) obrazuje relację między siłą zmiany ekspresji (oś X: logFC) a jej istotnością statystyczną (oś Y: -log10 FDR). Czerwone punkty reprezentują geny ulegające zwiększeniu ekspresji (zostały aktywowane) w obecności czynnika EV. Punkty niebieskie oznaczają geny, których ekspresja została istotnie obniżona (została wyciszona) względem kontroli.

Heatmapa (Rysunek 12) przedstawia znormalizowane wartości ekspresji (skalowanie Z-score), co pozwala na śledzenie trendów w punktach czasowych 0h, 12h i 24h. Gradient kolorystyczny oparty na przejściu od barwy niebieskiej (niska ekspresja) do czerwonej (wysoka ekspresja) umożliwia wizualną identyfikację genów o zbliżonym profilu reakcji na badane warunki.

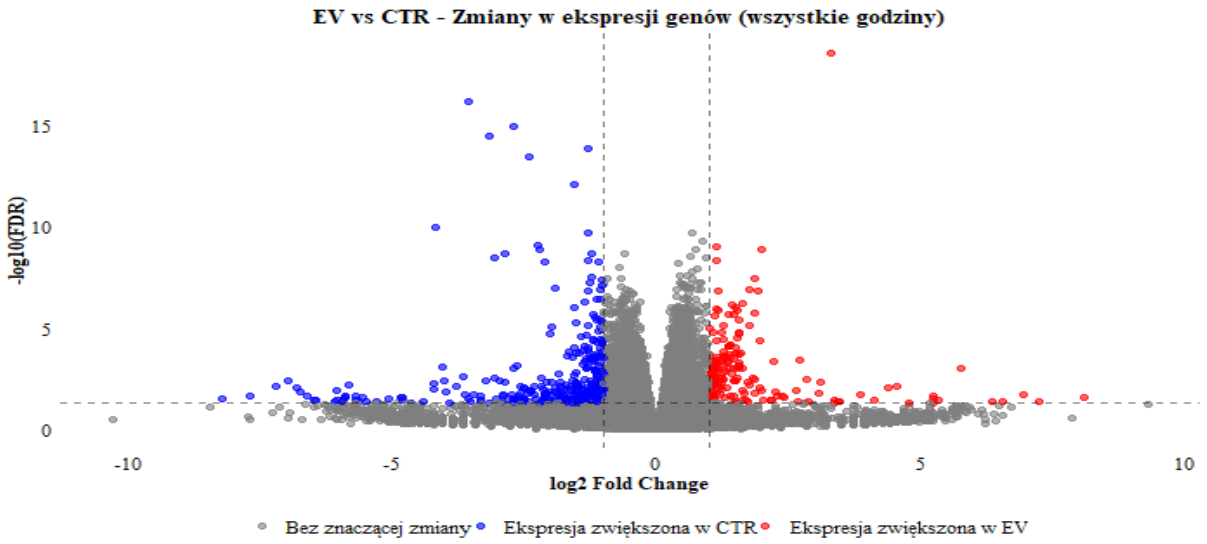

Figure 11 Analysis 4 Volcano Plot

Rysunek 11 Analiza 4 Volcano Plot

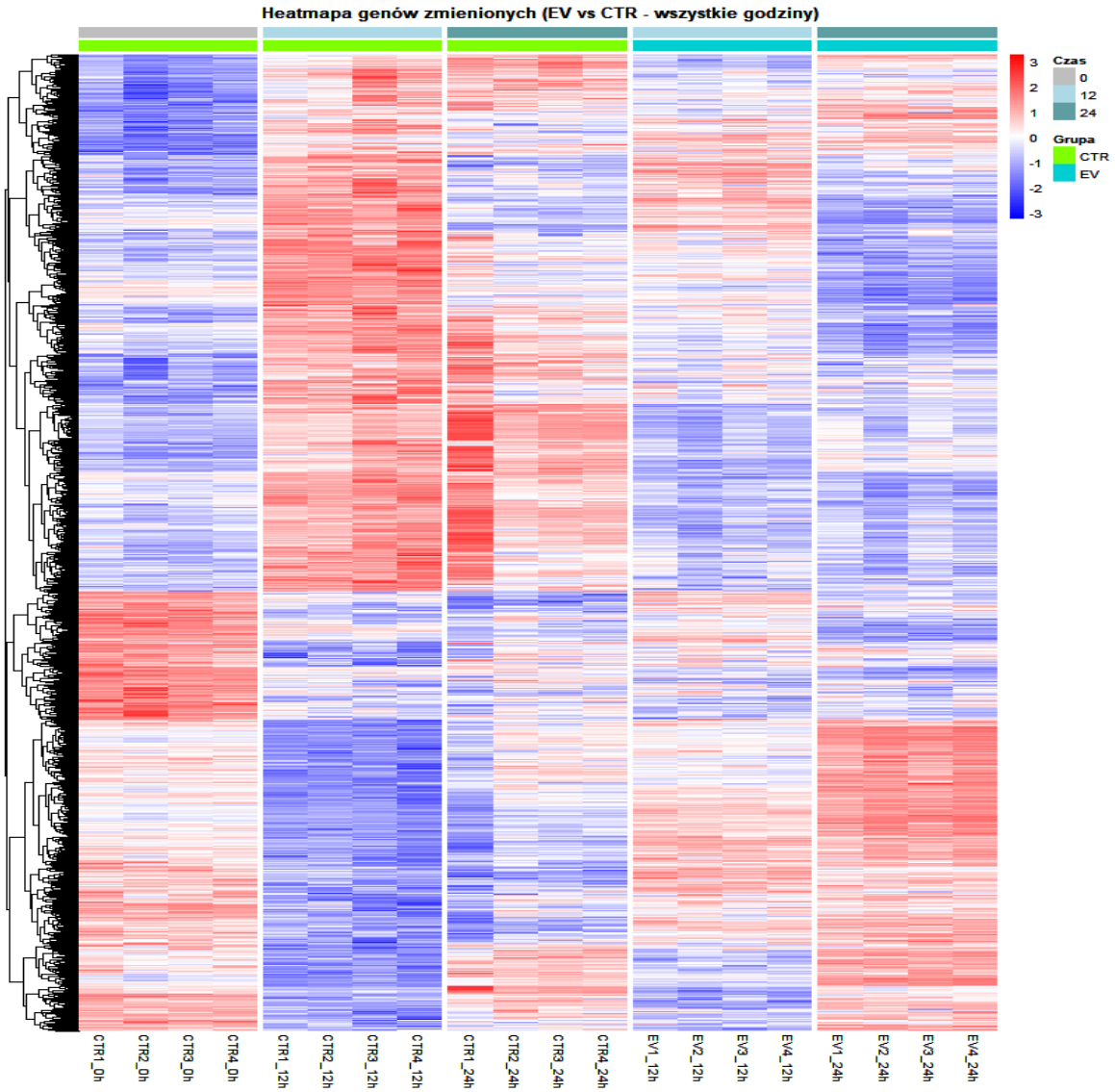

Figure 12 Analysis 4 Heatmap

Rysunek 12 Analiza 4 Heatmap

Analiza 4 (Dynamika zmian ekspresji genów w czasie) pozwoliła wyjść poza statyczne porównania punktowe i ukazała dynamikę procesu biologicznego. Wykazuje ona, jak badany czynnik (EV) modyfikuje ekspresję genów w czasie, co lepiej odzwierciedla rzeczywisty przebieg reakcji komórkowej w hodowli.

Analiza 5 - Grupa kontrolna CTR

Aby oszacować skalę zmian genetycznych wynikających wyłącznie z naturalnych procesów zachodzących w komórkach, przeprowadzono analizę porównawczą wewnątrz grupy kontrolnej (CTR). Analiza zbiorów na diagramie venna (Rysunek 13) ukazuje relacje pomiędzy trzema zestawami genów, których ekspresja uległa istotnej zmianie (FDR < 0.05) w porównaniach parowych: 12h vs 0h, 24h vs 0h oraz 24h vs 12h. Część wspólna wszystkich

trzech zbiorów obejmuje 90 genów. Są to zmienne, których poziom ekspresji ulega ciągłej modyfikacji na każdym etapie trwania eksperymentu, niezależnie od przedziału czasowego.
Zidentyfikowano liczne grupy genów unikalnych dla konkretnych porównań (np. 1073 geny zmieniające się tylko w perspektywie 24h vs 0h, ale nieistotne w krótszych okresach).
Analiza ta dowodzi, że hodowla komórkowa jest układem dynamicznym. Wykrycie genów zmieniających się w samej kontroli potwierdza konieczność stosowania modelu interakcji (Analiza 3), aby oddzielić szum biologiczny od właściwego sygnału wywołanego przez pęcherzyki EV.

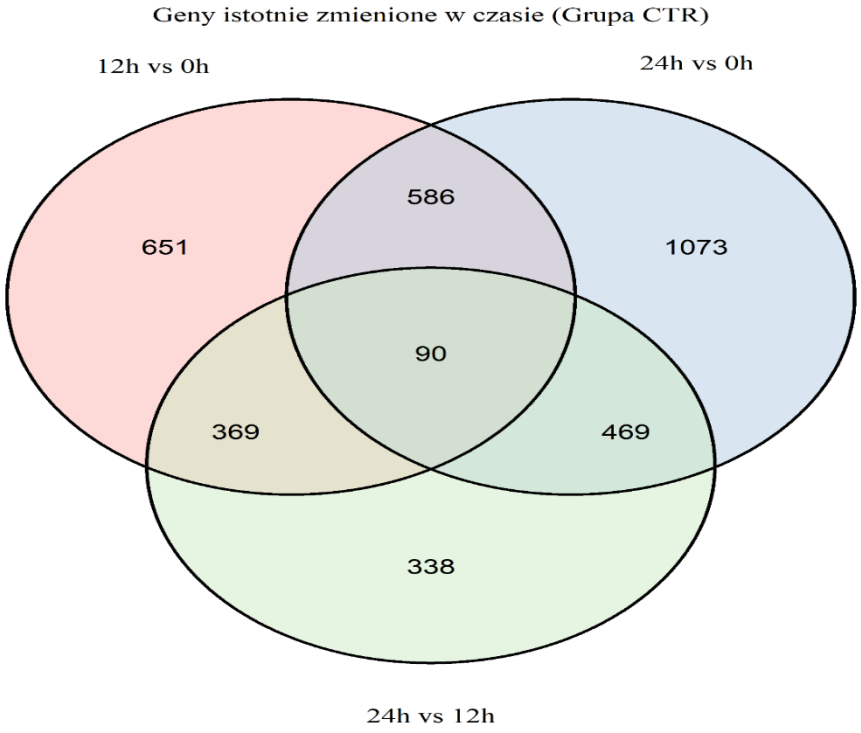

Figure 13 Analysis 5 Venn Diagram

Rysunek 13 Analiza 5 Diagram Venna

Wykres (Rysunek 14) obrazuje wczesne zmiany ekspresji 1696 (liczba widoczna w nagłówku wykresu) genów zachodzące w grupie kontrolnej. Liczba ta oznacza dynamiczny start procesów zachodzących w komórkach. Klastrowanie wierszy ujawnia dwie główne grupy transkryptów: te, które ulegają samoistnej aktywacji po 12 godzinach (przejście z koloru niebieskiego na czerwony) oraz te, które są wyciszane (przejście z czerwonego na niebieski). Potwierdza to, że znacząca przebudowa transkryptomu następuje już w pierwszej dobie hodowli, niezależnie od badanego czynnika.

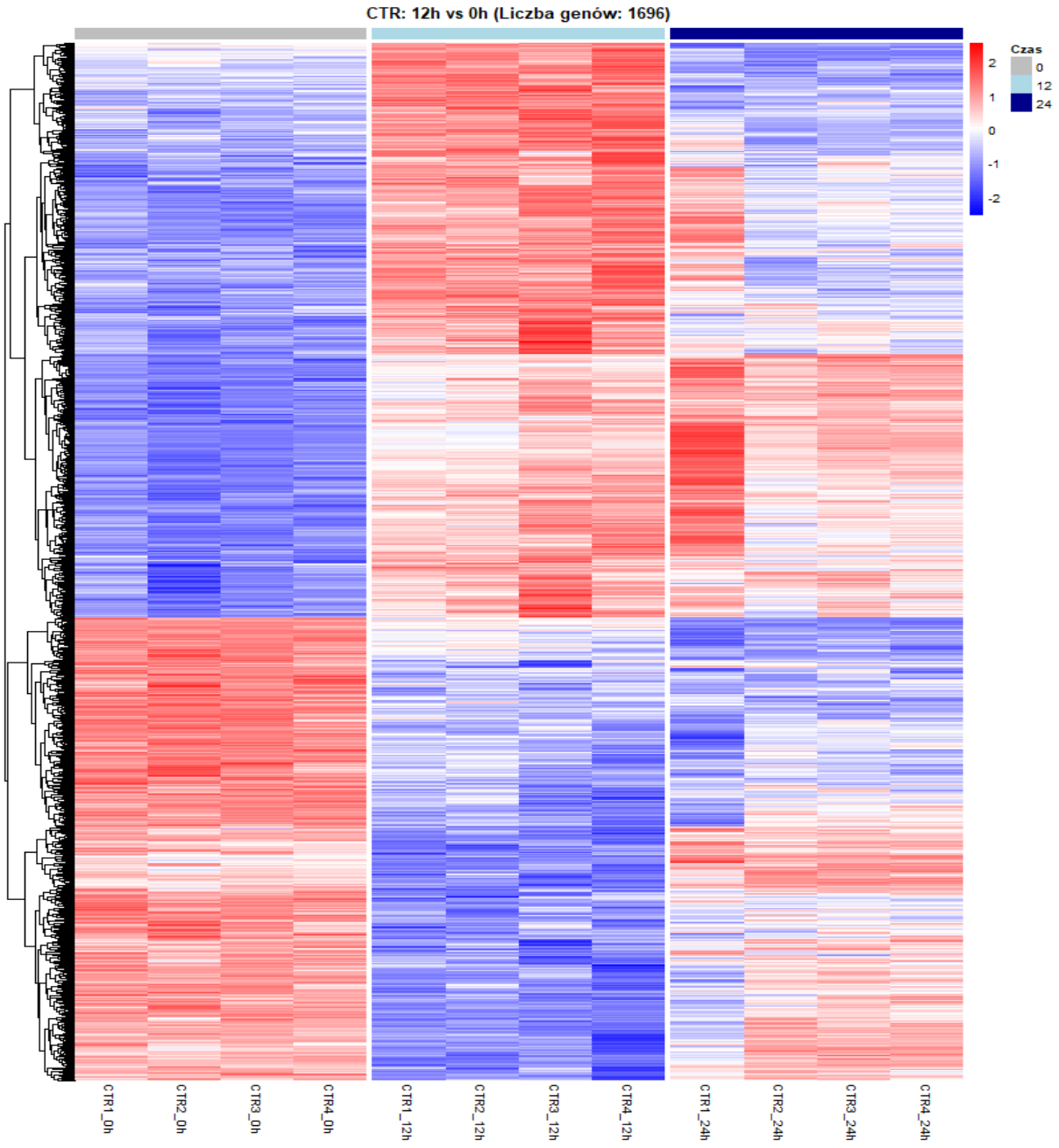

Figure 14 Analysis 5 Heatmap 12h vs 0h

Rysunek 14 Analiza 5 Heatmap 12h vs 0h

Wykres Volcano (Rysunek 15 Analiza 5 Wykres Volcano 12h vs 0h) przedstawia rozkład zmian ekspresji genów w grupie kontrolnej po 12 godzinach. Jest on uzupełnieniem wizualizacji danych widocznych na Rysunku 14 (Analiza 5 Heatmap 12h vs 0h). Oś X (log2 Fold Change) obrazuje kierunek i siłę zmiany. Punkty po prawej stronie (wartości dodatnie) oznaczają wzrost, a po lewej (wartości ujemne) spadek ekspresji względem czasu 0h trwania hodowli. Oś Y (-log10 FDR) wskazuje na istotność statystyczną wyniku. Im punkt jest położony wyżej, tym mniejsze prawdopodobieństwo błędu. Czerwone punkty odpowiadają za geny istotnie aktywowane (zwiększona ekspresja), natomiast niebieskie punkty to geny istotnie wyciszone

(zmniejszona ekspresja). Szare punkty poniżej przerywanych linii progowych oznaczają geny, które nie wykazały istotnych statystycznie zmian.

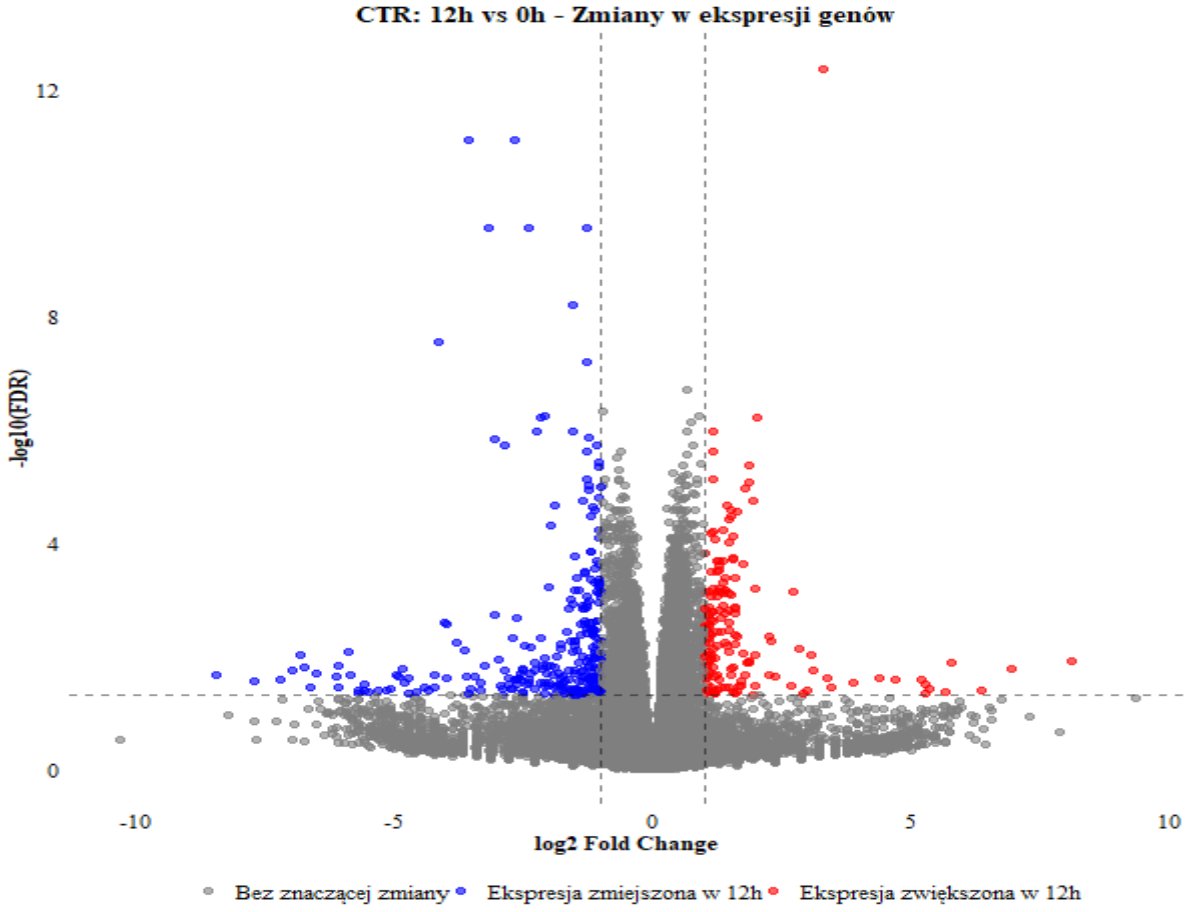

Figure 15 Analysis 5 Volcano Plot 12h vs 0h

Rysunek 15 Analiza 5 Wykres Volcano 12h vs 0h

Heatmapa (Rysunek 16) wizualizuje zestawienie 2218 genów, które zostały zakwalifikowane jako istotnie zmienione w porównaniu 24-godzinnym względem punktu startowego hodowli. Liczba ta, widoczna w nagłówku wykresu, wskazuje na aktywnie postępujące zmiany transkryptomu. Jest ona wyższa niż w pierwszym porównaniu (Rysunek 14 Analiza 5 Heatmap 12h vs 0h), gdzie zidentyfikowano 1696 genów. Informacja potwierdza postępujący charakter zmian. Analiza klastrów uwidacznia dwa przeciwstawne trendy: grupę genów, których aktywność drastycznie rośnie w 24. godzinie (górna część wykresu, intensywna czerwień) oraz grupę genów ulegających silnemu wyciszeniu (dolna część, kolor niebieski).

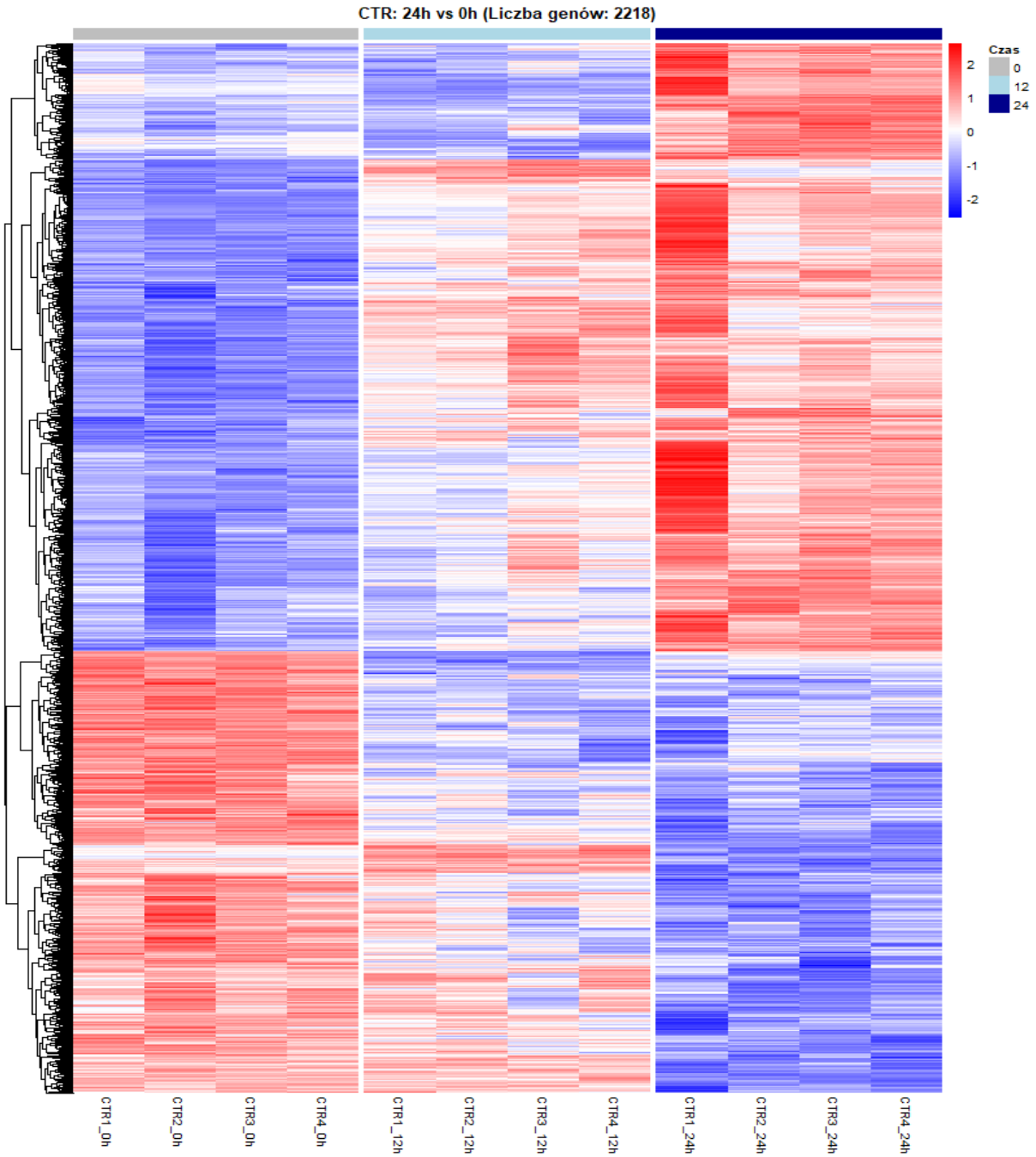

Figure 16 Analysis 5 Heatmap 24h vs 0h

Rysunek 16 Analiza 5 Heatmap 24h vs 0h

Wykres Volcano (Rysunek 17 Analiza 5 Wykres Volcano 24h vs 0h) ilustruje zmiany ekspresji genów po pełnej dobie hodowli (24h) względem punktu startowego. W porównaniu do okresu 12-godzinnego, obserwuje się tu zwiększoną liczbę punktów przekraczających progi istotności (linie przerywane), co łączy się z wynikiem widocznym na heatmapie (Rysunek 16 Analiza 5 Heatmap 24h vs 0h). Rozkład punktów wskazuje na pogłębiające się z czasem rozregulowanie i gwałtowne zmiany transkryptomu. Można zauważyć zarówno wzrost silnej ekspresji (czerwone punkty po prawej), jak i głębokie wyciszenie (niebieskie punkty po lewej).

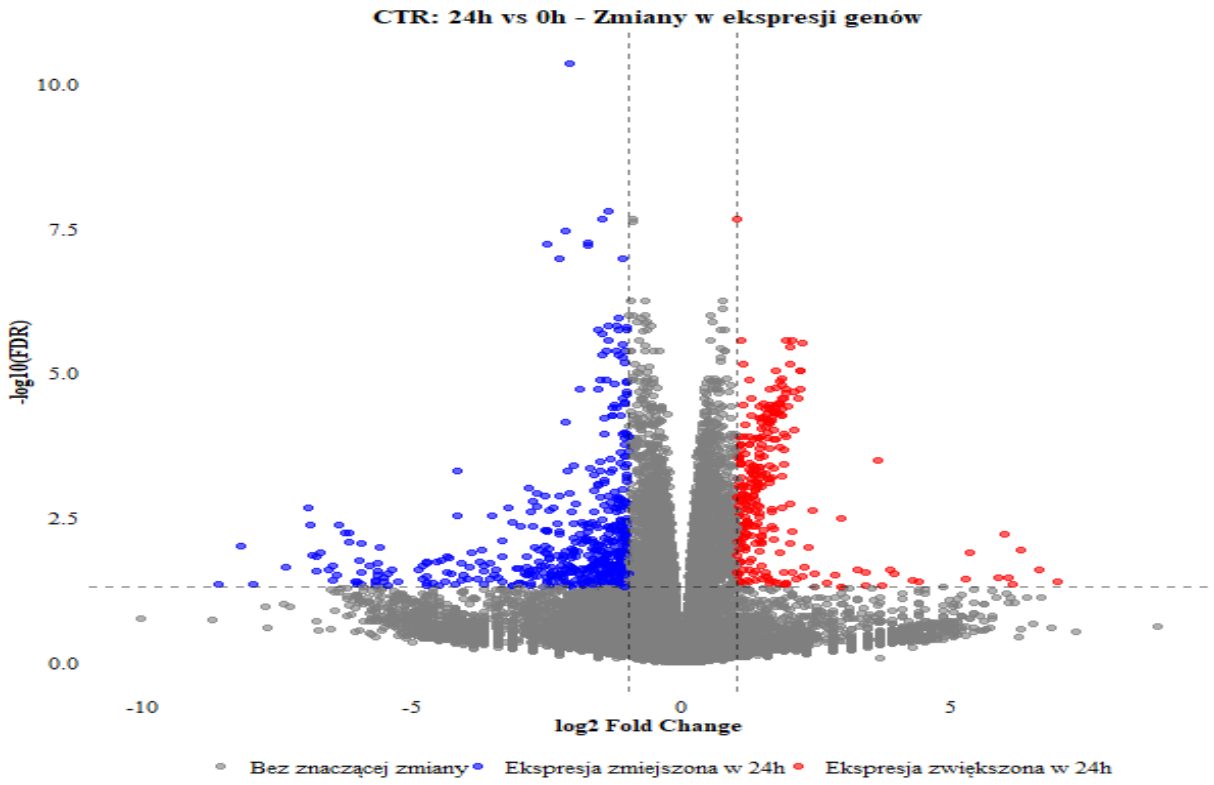

Figure 17 Analysis 5 Volcano Plot 24h vs 0h

Rysunek 17 Analiza 5 Wykres Volcano 24h vs 0h

Heatmapa (Rysunek 18 Analiza 5 Heatpmap 24h vs 12h) wizualizuje różnice w ekspresji 1266 genów pomiędzy środkowym czasem 12 godzin trwania hodowli a końcowym punktem czasowym (24h) w grupie kontrolnej. Wykres ten potwierdza informację, iż procesy adaptacyjne w komórkach nie kończą się w pierwszej części procesu, lecz trwają nadal. Dobrze widoczna zmiana kolorystyki między kolumnami reprezentującymi 12 i 24 godzinę wskazuje na grupę genów, których ekspresja ulega dalszej modyfikacji w późniejszej fazie hodowli. Profil genetyczny w 24 godzinie różni się znacząco od tego w 12 godzinie. Wynik ten potwierdza, że uwzględnienie czynnika czasu (jak w Analizie 3) było kluczowe. Bez tego zabiegu, naturalne zmiany fizjologiczne (starzenie) zachodzące w hodowli mogłyby zostać błędnie zinterpretowane jako efekt działania pęcherzyków EV.

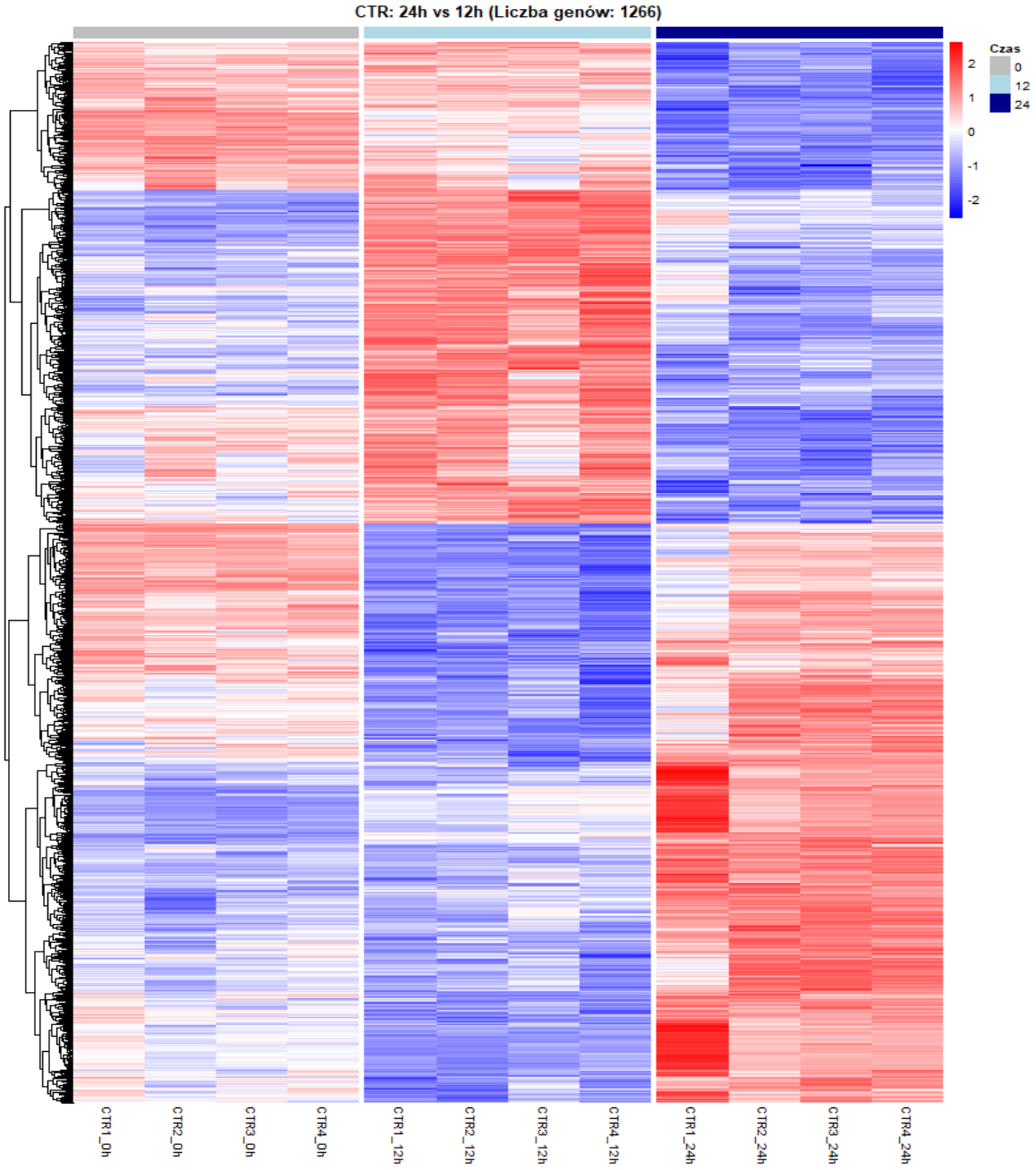

Figure 18 Analysis 5 Heatmap 24h vs 12h

Rysunek 18 Analiza 5 Heatmap 24h vs 12h

Wykres Volcano (Rysunek 19 Analiza 5 Wykres Volcano 24h vs 12h) dopełnia analizę różnic wewnątrz grupy kontrolnej. Zestawia ona końcowy etap hodowli (24h) z etapem pośrednim (12h). W porównaniu do wykresów odnoszących się do punktu zerowego, obserwuje się tu mniejszą liczbę punktów o skrajnych wartościach. Pozwala to na stwierdzenie, że dynamika zmian genetycznych znacząco zwalnia w drugiej dobie eksperymentu. Widoczna obecność genów istotnie nadekspresjonowanych (czerwone punkty) oraz wyciszonych (niebieskie punkty) dowodzi, że transkryptom komórek nie osiągnął jeszcze stanu stabilnego i nadal ulega przebudowie.

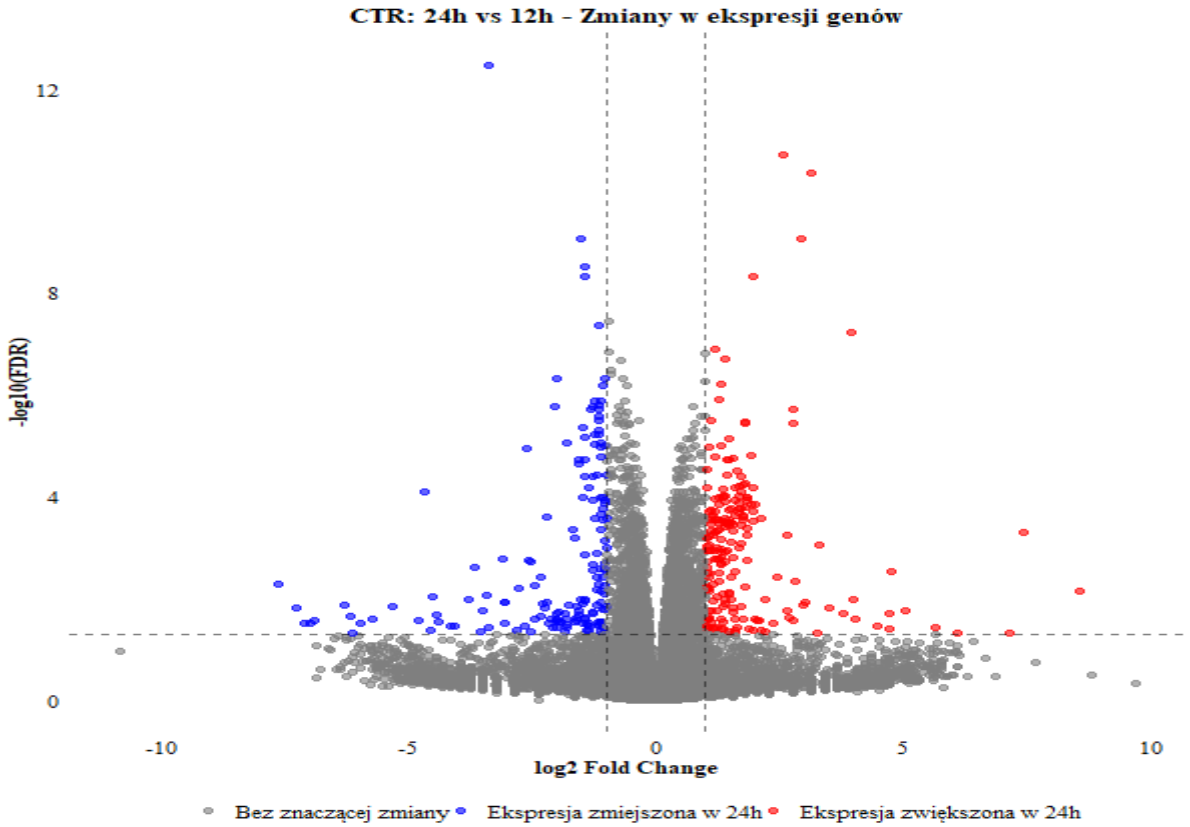

Figure 19 Analysis 5 Volcano Plot 24h vs 12h

Rysunek 19 Analiza 5 Wykres Volcano 24h vs 12h

Analiza trendu wyników modelu wieloczynnikowego

Wykres panelowy (Rysunek 20 Analiza trendu wyników modelu wieloczynnikowego) prezentuje indywidualne profile ekspresji 19 genów, które wykazały istotną statystycznie interakcję między traktowaniem a czasem. Każdy panel odpowiada pojedynczemu genowi, co umożliwia szczegółową analizę jego specyficznej reakcji. Oś X reprezentuje punkty czasowe (0h, 12h, 24h) natomiast oś Y przedstawia znormalizowany poziom ekspresji (LogCPM).
Punkty pomarańczowe, brązowe oraz zielone oznaczają grupę kontrolną (CTR), a fioletowe i turkusowe grupę badaną (EV) w różnych grupach czasowych. Wizualizacja ujawnia cechy wspólne dla wyselekcjonowanych genów:

- W punkcie startowym (0h) oraz często w 12 godzinie, poziomy ekspresji dla obu grup są zbliżone (punkty nakładają się lub leżą blisko siebie).
- Wyraźne rozjeżdżanie się trendów następuje zazwyczaj dopiero w 24 godzinie.

Potwierdza to, że efekt działania pęcherzyków EV nie jest natychmiastowy, lecz narasta w czasie, wpływając na aktywność genów w późniejszej fazie eksperymentu.

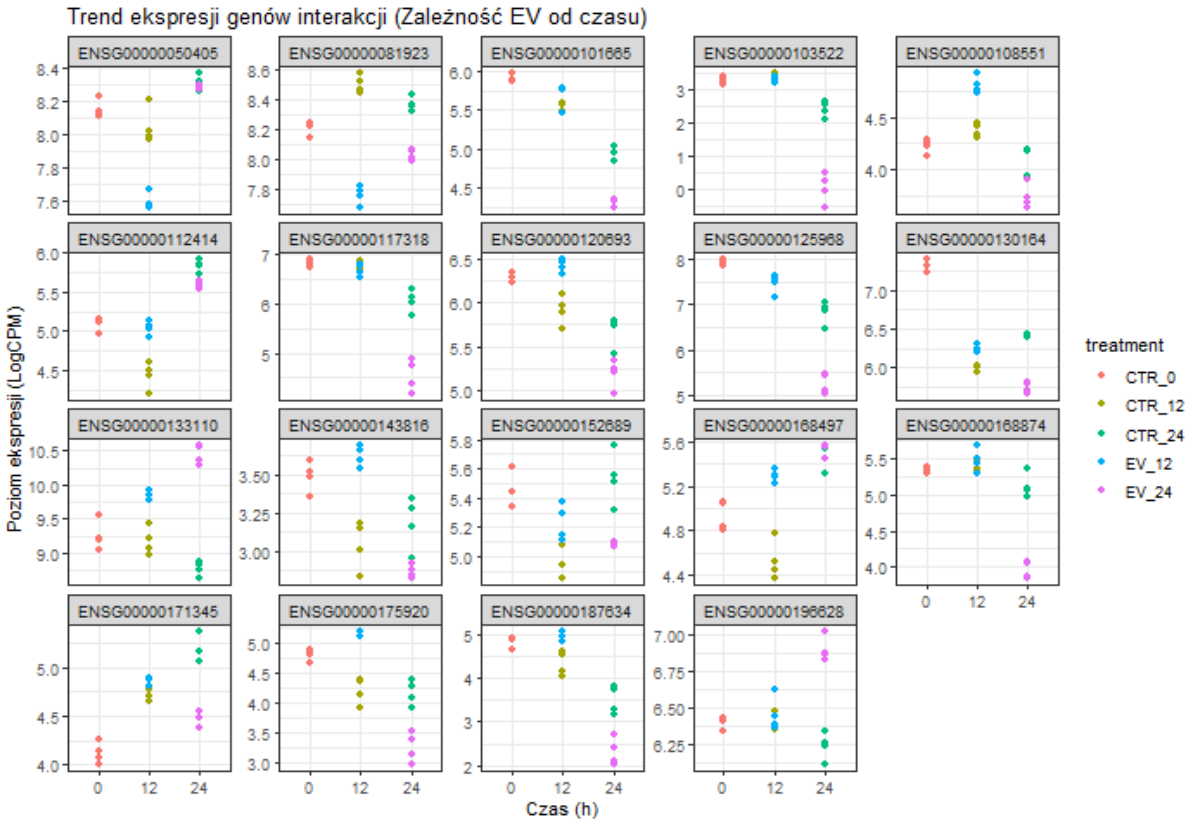

Figure 20 Trend analysis of multifactorial model results

Rysunek 20 Analiza trendu wyników modelu wieloczynnikowego

Selekcja genów o najwyższym współczynniku zróżnicowania ekspresji

Etap ten odpowiada za zilustrowanie najsilniejszych trendów zmian zidentyfikowanych w modelu wieloczynnikowym. Wybrano trzy geny o najwyższym zróżnicowaniu ekspresji:

- ENSG00000103522 - IL21R,
- ENSG00000133110 - POSTN
- ENSG00000196628 - TCF4

Wykresy punktowe przedstawiają poziom ich ekspresji w funkcji czasu z podziałem na grupę kontrolną (kolor zielony) oraz grupę badaną EV (kolor fioletowy). Dla wszystkich trzech genów poziomy ekspresji w punkcie początkowym są zbliżone dla obu grup, co potwierdza brak istotnych różnic przed wprowadzeniem czynnika eksperymentalnego. Wraz z upływem czasu (w 12, a szczególnie w 24 godzinie) obserwuje się wyraźne rozdzielenie punktów reprezentujących poszczególne grupy. Świadczy to o tym, że wpływ pęcherzyków zewnątrzkomórkowych na ekspresję tych konkretnych genów nie jest stały, lecz następuje w czasie.

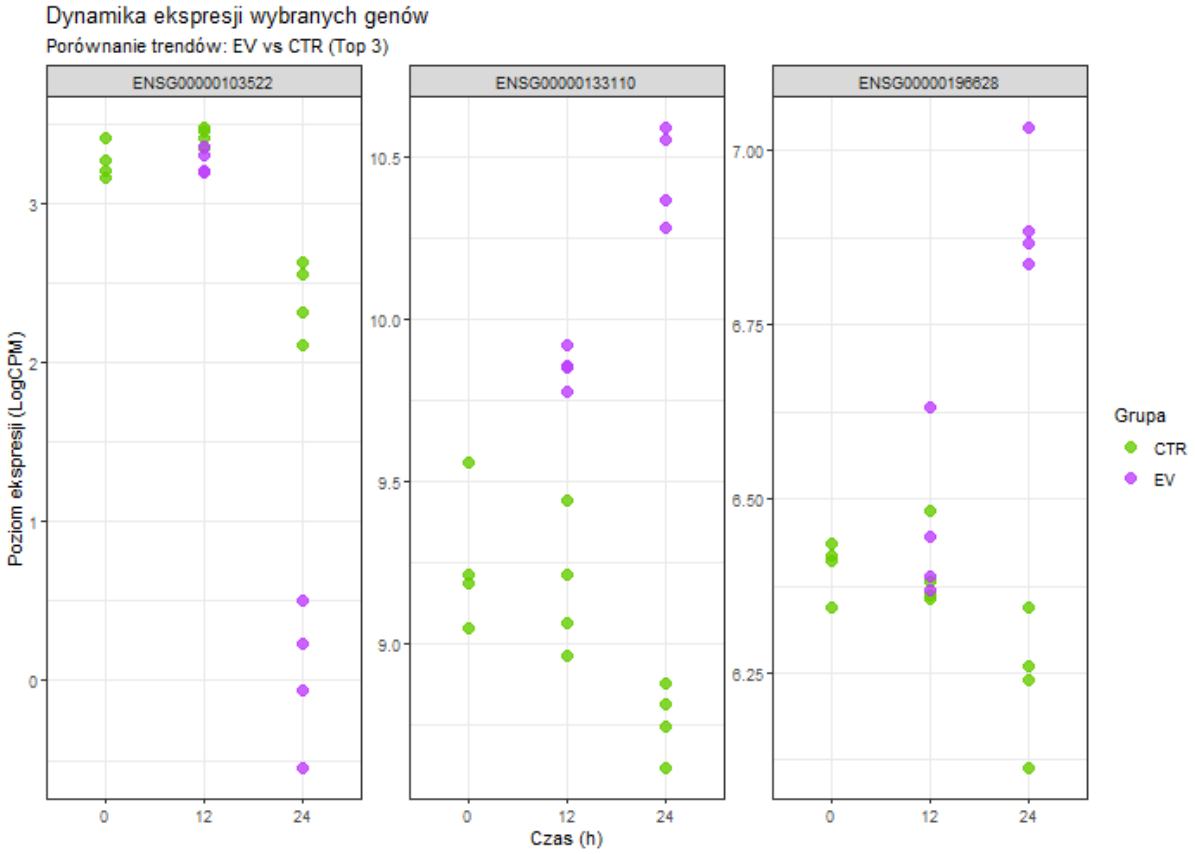

Figure 21 Genes with the highest differential expression levels

Rysunek 21 Geny o najwyższym współczynniku zróżnicowania ekspresji

Geny unikatowe dla analizy wieloczynnikowej

Wykres ukazuje profile ekspresji dwóch genów:

- ENSG00000108551 - RASD1
- ENSG00000196628 - TCF4

Geny te zostały zidentyfikowane w Analizie 3 jako zmienne unikalne dla efektu interakcji. Oznacza to, że ich zmienność nie wynika z samego upływu czasu ani obecności czynnika, lecz ze specyficznego połączenia obu tych warunków. W przypadku obu genów obserwuje się nieliniową dynamikę zmian. Linie trendu dla grupy kontrolnej (zielony) i badanej (fioletowy) przecinają się lub gwałtownie zmieniają kierunek między 12 a 24 godziną. Taki przebieg ekspresji jest trudny do uchwycenia w standardowych testach parowych. Wyizolowanie tych genów potwierdza, że zastosowany model matematyczny skutecznie wykrył subtelne, ale biologicznie istotne sygnały, które zostałyby utracone przy zastosowaniu prostszych metod analitycznych.

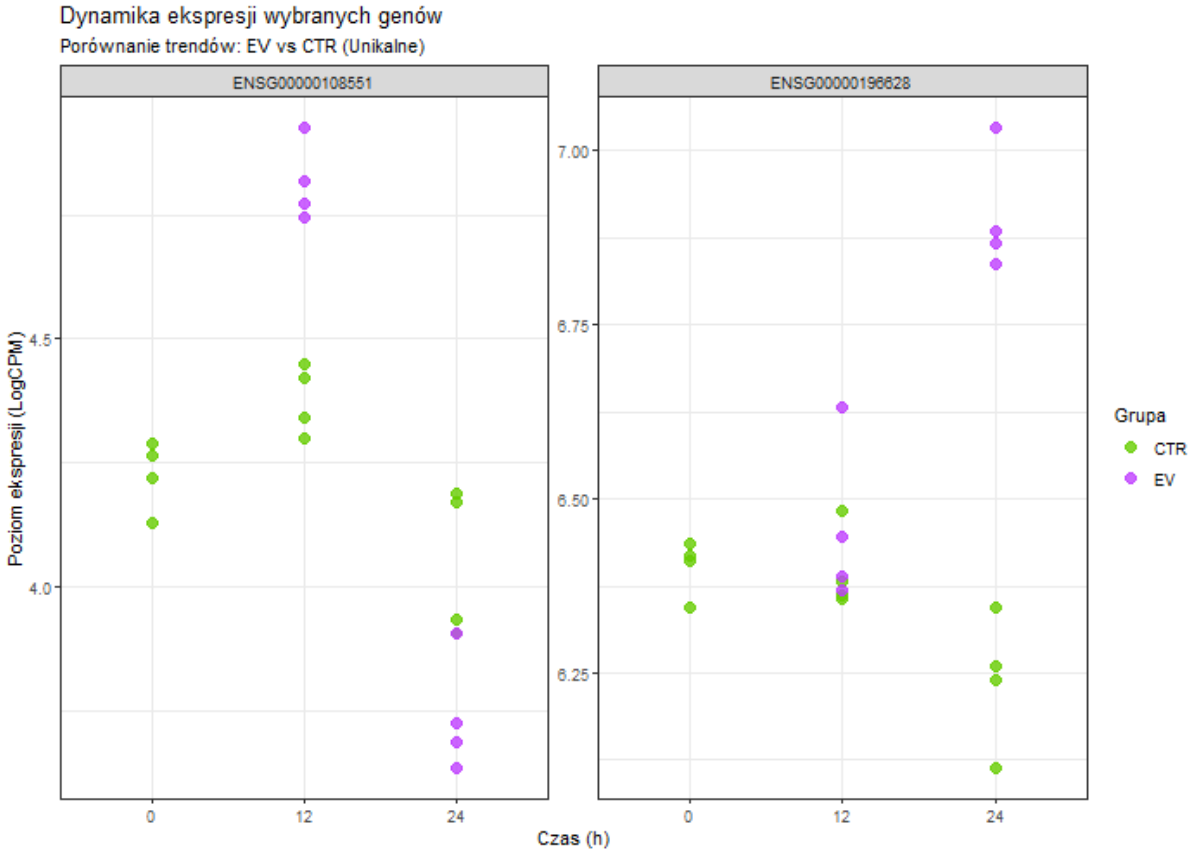

Figure 22 Genes unique to multifactorial analysis

Rysunek 22 Geny unikatowe dla analizy wieloczynnikowej

Dyskusja

Głównym celem niniejszej pracy było zaprojektowanie i implementacja potoku przetwarzania danych (data pipeline), który poradzi sobie z problemem „Big Data" w skali mikro, czyli z ogromną liczbą zmiennych (ponad 60 tysięcy genów) przy małej liczbie próbek. Kluczowym wyzwaniem analitycznym było odróżnienie sygnału właściwego (wpływ pęcherzyków EV na grupę kontrolną) od szumu (naturalne zmiany w czasie – starzenie hodowli).

Wybór modelu statystycznego - Analiza jednoczynnikowa a wieloczynnikowa:

W początkowej fazie projektu zastosowano standardowe podejście porównania par, realizowane w Analizie 1 i 2 (Porównanie warunków EV i CTR w poszczególnych punktach czasowych). Polegało ono na prostym porównaniu grupy badanej z kontrolną w konkretnych punktach czasowych. Jest to podejście, często stosowane w prostych analizach A/B. W przypadku danych szeregów czasowych okazało się one niewystarczające. Jak zauważono w badaniach nad narzędziami RNA-seq wykonanych przez Daniela Spies i innych [5], proste porównania statyczne w danych dynamicznych prowadzą do utraty informacji o trendzie.

Potwierdziły to wyniki Analizy 5 (Grupa kontrolna CTR) z niniejszej pracy. Dowiodła ona, że w samej grupie kontrolnej, gdzie nie zastosowano żadnego czynnika zewnętrznego aż 2218 cech (genów) zmieniło swoją wartość po 24 godzinach. Z perspektywy analizy danych, czas stał się zmienną zakłócającą. Poprzestanie na prostych porównaniach takich jak analiza 1 i 2 spowodowałoby błędne zaklasyfikowanie naturalnych zmian jako efektu działania czynnika badanego. Rozwiązaniem tego problemu było zastosowanie w Analizie 3 uogólnionego modelu liniowego (GLM) z interakcją (condition * time). Jest to podejście rekomendowane w literaturze statystycznej, m.in. przez twórców pakietu edgeR [7]. Model ten działa jak zaawansowany filtr danych. Matematycznie odejmuje zmienność wynikającą z upływu czasu (tło) od zmienności całkowitej, pozostawiając jedynie te cechy, które reagują na badany czynnik w sposób nieliniowy.

Zadaniem napisanego skryptu była redukcja wymiarowości problemu. Na wejściu algorytm otrzymał macierz o wymiarach 63 241 wierszy (genów). Dzięki zastosowaniu rygorystycznych filtrów statystycznych (FDR < 0.01) w modelu interakcji, system automatycznie wyselekcjonował 19 kluczowych zmiennych. Z punktu widzenia data science, jest to klasyczny proces wybierania najbardziej istotnych zmiennych spośród dostępnych danych (Feature Selection). Uzyskany wynik (19 z 63 241 genów) świadczy o wysokiej selektywności modelu. Wykresy trendu (Rysunek 20) potwierdzają, że algorytm zadziałał poprawnie. Wybrane zostały tylko te przypadki, gdzie krzywe dla grupy badanej i kontrolnej wyraźnie się rozjeżdżają. Wykresy dla genów unikalnych (Rysunek 22) pokazują z kolei, że model potrafi wykryć nawet skomplikowane wzorce, których nie wykryłaby prosta analiza korelacji.

Realizacja tego projektu pozwoliła na wyciągnięcie następujących wniosków:

- Surowe dane biologiczne są silnie zaszumione. Bez zastosowania normalizacji (TMM) i transformacji logarytmicznej, wizualizacje takie jak PCA (Rysunek 1) byłyby nieczytelne co wskazuje na konieczność wykonania normalizacji danych.
- Stworzony skrypt w języku R pozwala na przetworzenie dużych ilości danych w kilkanaście minut. Daje to ogromną przewagę nad ręczną analizą w arkuszach kalkulacyjnych (takich jak Excel), która przy tej ilości danych byłaby niemożliwa.
- Jak podkreślają Gutman i Goldmeier [11], rolą analityka nie jest tylko uruchomienie kodu, ale zrozumienie modelu. Fakt, że wyselekcjonowane geny mają sens biologiczny, jest najlepszym dowodem na to, że napisany kod i dobrany model matematyczny działają prawidłowo.

Spis rysunków

List of Figures

Bibliografia

[1] Dhrithi Deshpande i inni, „RNA-seq data science: From raw data to effective interpretation” https://www.frontiersin.org/journals/genetics/articles/10.3389/fgene.2023.997383/full

[2] Kimberly R. Kukurba, Stephen B. Montgomery „RNA Sequencing and Analysis” https://pmc.ncbi.nlm.nih.gov/articles/PMC4863231/

[3] Dominick Sinicropi i inni, „Whole Transcriptome RNA-Seq Analysis of Breast Cancer Recurrence Risk Using Formalin-Fixed Paraffin-Embedded Tumor Tissue” https://pmc.ncbi.nlm.nih.gov/articles/PMC3396611/

[4] Sunghee Oh, Seongho Song, Nupur Dasgupta, Gregory Grabowski „The analytical landscape of static and temporal dynamics in transcriptome data” https://pmc.ncbi.nlm.nih.gov/articles/PMC3929947/

[5] Daniel Spies, Peter F. Renz, Tobias A. Beyer, Constance Ciaudo „Comparative analysis of differential gene expression tools for RNA sequencing time course data” https://pmc.ncbi.nlm.nih.gov/articles/PMC6357553/

[6] DESeq2 https://bioconductor.org/packages/release/bioc/manuals/DESeq2/man/DESeq2.pdf

[7] edgeR https://bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/intro.html

[8] biomaRt
https://bioconductor.org/packages/release/bioc/html/biomaRt.html
[9] Ian T. Jolliffe, Jorge Cadima „Principal component analysis: a review and recent developments”
https://pmc.ncbi.nlm.nih.gov/articles/PMC4792409/
[10] Jordan Gldmeier „Mistrz analizy danych, od danych do wiedzy – wydanie II”, Wydawnictwo Helion.
[11] Gutman A. J., Goldmeier J., „Analityk danych. Przewodnik po data science, statystyce i uczeniu maszynowym”, Wydawnictwo Helion.
[12] Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika
http://cda.psych.uiuc.edu/psychometrika_highly_cited_articles/kruskal_1964a.pdf