Test assignment

Bence Szikora

2025-02-11

Task description

Write a script that perform the following tasks:

- Normalize raw gene counts by calculating RPKM and TPM values
- Plot density curves before and after the normalization with the following requirements:
  - Logarithmize the gene counts
  - Draw a curve for each sample with different color
  - Add title and labels to the axes
  - Add a legend to describe the meaning of the colors
  - Prefer using basic R graphical functions

Background

To eliminate technical biases in sequenced data, such as sequencing depth (deeper sequencing depth produces more read counts for one gene) and gene length (longer gene length produces more read counts at the same sequencing level), normalization of gene expression measurements is required.

RPKM (Reads Per Kilobase per Million mapped reads) was made for single-end RNA-seq, where every read corresponded to a single fragment that was sequenced.We divide the number of fragments of a gene by the total sequencing depth, and the ratio is divided by the gene length.

TPM (transcripts per kilobase million) is very much like RPKM, but the only difference is that at first, normalize for gene length, and later normalize for sequencing depth. TPM is a more accurate statistic when calculating gene expression comparisons across samples.

Based on: https://www.novogene.com/eu-en/resources/blog/how-to-choose-normalization-methods-tpm-rpkm-fpkm-for-mrna-expression/

Script

The files were unzipped manually, and the working directory was set. The following packages were used for the script: dplyr, tidyr

Generate TPM and RPKM values

```
# Load necessary libraries
library(dplyr)
```

```
##
## Kapcsolódás csomaghoz: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

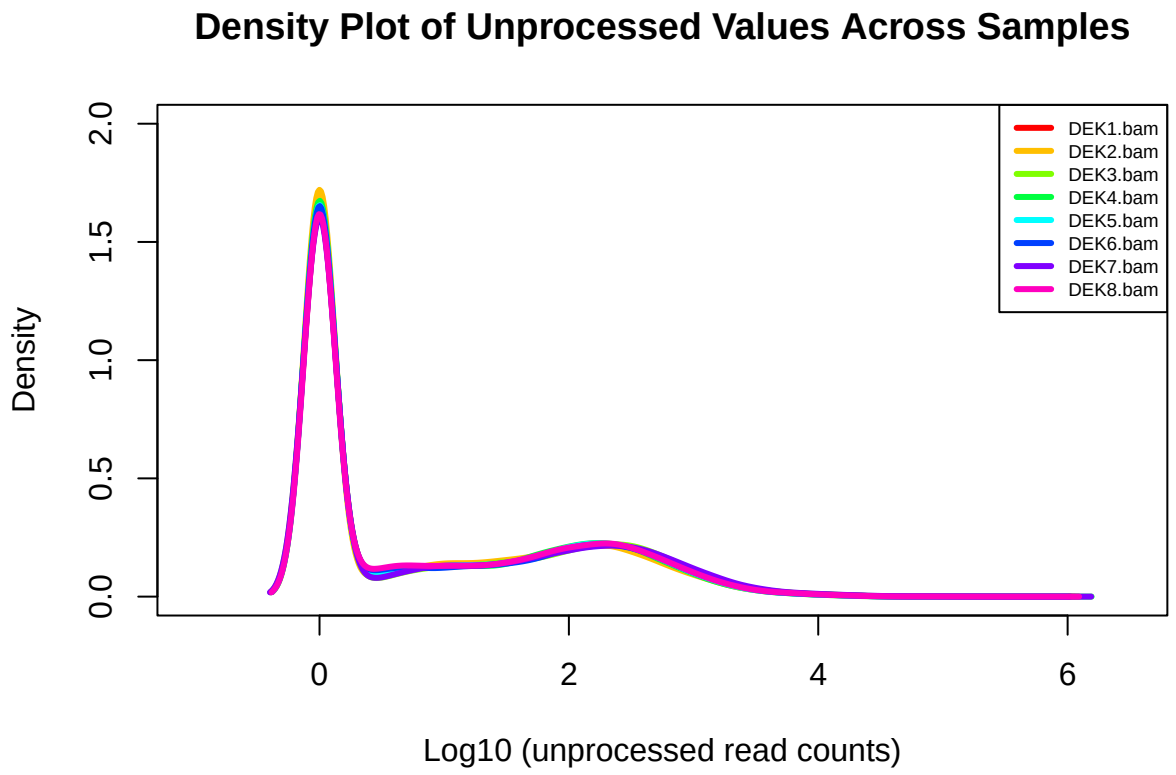
```
library(tidyr)
```

```
#read the file
file<-read.table(file = "decorin__feature_counts_20200617_1vb.txt", header = TRUE)

#use only the necessary columns
data<- file%>%
        select(1,(ncol(file)-8):ncol(file))

# Convert data to long format
data_long <- data %>%
        pivot_longer(cols = starts_with("DEK"), names_to = "Sample", values_to = "Counts")

# Compute TPM for each sample
data_tpm <- data_long %>%
        group_by(Sample) %>%
        mutate(RPK = Counts / (Length / 1000)) %>%  # Reads per kilobase
        mutate(TPM = (RPK / sum(RPK)) * 1e6) %>%   # Normalize to TPM
        select(Geneid, Sample, TPM) %>%
        pivot_wider(names_from = Sample, values_from = TPM)  # Convert back to wide format

#save the TPM data
write.csv(data_tpm, "skill_survey_tpm_20250211_1530_1SzB.csv", row.names = FALSE)

# Compute RPKM for each sample
data_rpkm <- data_long %>%
        group_by(Sample) %>%
        mutate(RPM = (Counts / sum(Counts)) * 1e6) %>%   # Reas per million
        mutate(RPKM = RPM / (Length / 1000)) %>%  # Normalize to RPKM
        select(Geneid, Sample, RPKM) %>%
        pivot_wider(names_from = Sample, values_from = RPKM)  # Convert back to wide format

#save the RPKM data
write.csv(data_rpkm, "skill_survey_rpkm_20250211_1545_1SzB.csv", row.names = FALSE)
```

Generate the plots

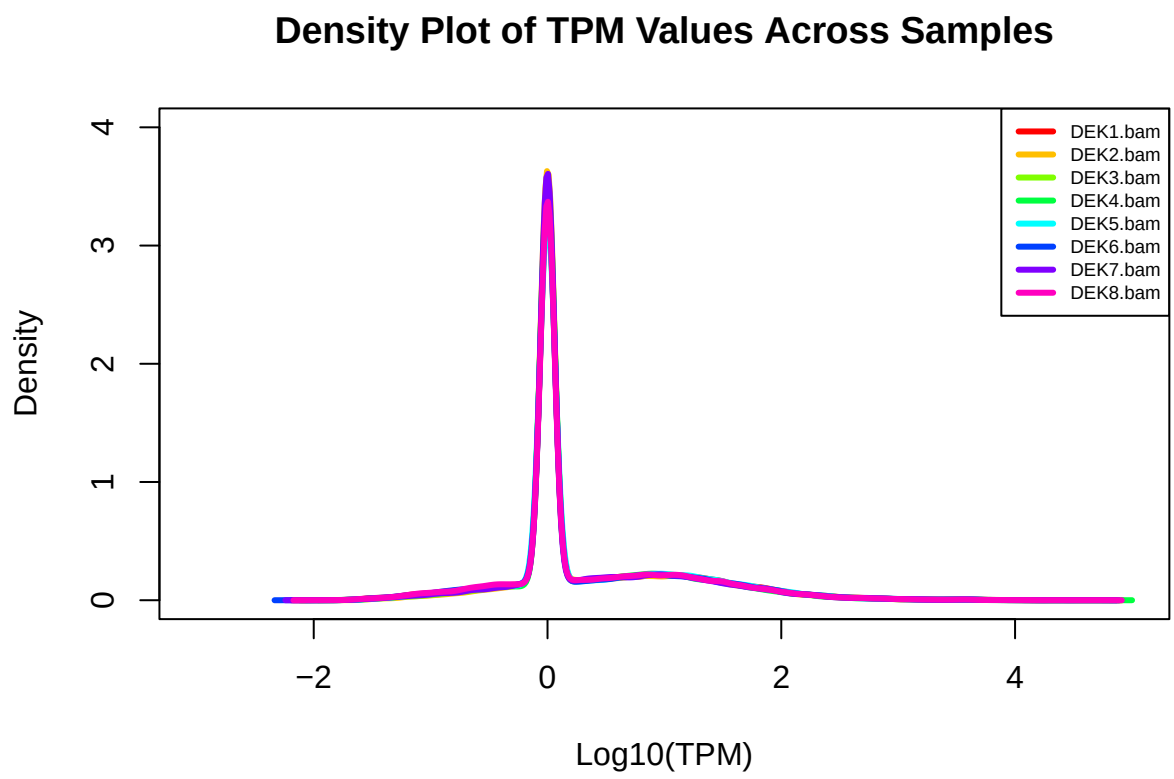
Plot for the unprocessed gene count:

```r
# Define colors for each sample
sample_colors <- rainbow(8)

#Logarithmize the unprocessed data
data_unp<- data[-2] #the len of the genes is not needed
data_unp_log <- data_unp %>%
        mutate(across(-Geneid, ~ ifelse(. == 0, 0, log10(.))))

#plot the unprocessed data
#check the unprocessed_log data

#summary(data_unp_log)

# Open a plot without data (to overlay curves later)
plot(NULL, xlim=c(-1,6.5), ylim=c(0, 2),
       xlab="Log10 (unprocessed read counts)", ylab="Density",
       main="Density Plot of Unprocessed Values Across Samples")

# Loop through each sample and plot density curve
for (i in 2:(9)) {
        lines(density(data_unp_log[[i]]), col = sample_colors[i-1], lwd = 3)
        }

# Add legend
legend("topright", legend=colnames(data_unp_log)[2:9], col=sample_colors, lwd=3, cex=0.6)
```



Plot for the TPM data:

```
#Logarithmize the TPM data
data_tpm_log <- data_tpm %>%
        mutate(across(-Geneid, ~ ifelse(. == 0, 0, log10(.))))

#plot the log_tpm values
#check the log_tpm data

#summary(data_tpm_log)

# Open a plot without data (to overlay curves later)
plot(NULL, xlim=c(-3,5), ylim=c(0, 4),
     xlab="Log10(TPM)", ylab="Density",
     main="Density Plot of TPM Values Across Samples")
# Loop through each sample and plot density curve

for (i in 2:(9)) {
        lines(density(data_tpm_log[[i]]), col = sample_colors[i-1], lwd = 3)
}

# Add legend
legend("topright", legend=colnames(data_tpm_log)[2:9], col=sample_colors, lwd=3, cex=0.6)
```



Plot for the RPKM data:

```r
#Logarithmize the RPKM data
data_rpkm_log <- data_rpkm %>%
        mutate(across(-Geneid, ~ ifelse(. == 0, 0, log10(.))))

#plot the log_rpkm values
#check the log_rpkm data

#summary(data_rpkm_log)

# Open a plot without data (to overlay curves later)
plot(NULL, xlim=c(-3,5), ylim=c(0, 4.2),
     xlab="Log10(RPKM)", ylab="Density",
     main="Density Plot of RPKM Values Across Samples")
# Loop through each sample and plot density curve

for (i in 2:(9)) {
        lines(density(data_rpkm_log[[i]]), col = sample_colors[i-1], lwd = 3)
}

# Add legend
legend("topright", legend=colnames(data_rpkm_log)[2:9], col=sample_colors, lwd=3, cex=0.6)
```