Analyse comparative des méthodes de normalisation RNA-Seq ciblé - SLA

Mickael Coquerelle

03 juillet 2025

Avant-propos et objectifs

Ce rapport sert de synthèse a mes différentes investigations et scripts/essais que j'ai pu effectuer en R, pour générer/évaluer mes données de comptages et surtout à mes investigations sur les méthodes de normalisation pour faire de l'analyse quantitative dans le périmètre du RNA-Seq ciblé sur le panel de 56 gènes et tout ce que cela implique en terme de contraintes et de limites, comme détaillé dans mes anciennes réalisations.

Ici je traite de **TPM** (intègre la longueur des genes), **TMM** (concu au départ pour le transcriptome complet) et **RLE** (hypothèse d'une médiane stable). Je tiens à rappeler également que les approches classiques dans notre contexte de faible nombre de gènes sont relativements fragiles (cf. mon travail bibliographique sur les méthodes de normalisation).

Préparation/mise en forme du dataset

```
#  Charger les librairies nécessaires.
#  edgeR : pour calcNormFactors (TMM, RLE) + cpm()
#  purrr : fusion propre des dataframes
library(readr)      # Import TSV
library(dplyr)      # Manipulations de données
library(tidyr)      # pivot_longer / separate
library(ggplot2)    # Visualisations
library(edgeR)      # Normalisation TMM & RLE
library(tibble)     # rownames_to_column
library(purrr)      # reduce pour fusion
library(broom)      # tidy() sur modèles statistiques
library(kableExtra) # Jolis tableaux LaTeX
```

Import/préparation de la matrice de comptages

```
# Lecture du fichier de comptages généré par featureCounts (-f).
# Colonnes attendues : Geneid | gene_name | Chr | Length | échantillons...

tableau_comptages_bruts <- read_tsv("~/Final_counts_56genes.tsv", show_col_types = FALSE)

# Renommage pour uniformiser la colonne gene_name en nom_gene
tableau_comptages_bruts <- tableau_comptages_bruts %>% rename(nom_gene = gene_name, Longueur = Lengt
```

```r
# Passage au format long pour faciliter les transformations ultérieures.
comptages_long <- tableau_comptages_bruts %>%
  pivot_longer(
    cols      = -c(Geneid, nom_gene, Chr, Longueur),
    names_to  = "ID_complet",
    values_to = "lectures_brutes"
  ) %>%
  separate(
    col     = "ID_complet",
    into    = c("run", "echantillon", "condition"),
    sep     = "-",
    extra   = "merge",
    remove  = TRUE
  )
```

Analyse exploratoires des données brutes

```r
# Somme des lectures par échantillon, vérification de la taille des lib
taille_bibliotheque <- comptages_long |> group_by(echantillon) |> summarise(somme_lectures =sum(lect
print(taille_bibliotheque)
```

```
## # A tibble: 88 x 2
##    echantillon somme_lectures
##    <chr>                <dbl>
##  1 C01P092            7181393
##  2 C01P269            3646210
##  3 C01P276            3296245
##  4 C01P114            2849830
##  5 C01P035            2666206
##  6 C01P003            2377025
##  7 C01P051            2287024
##  8 C01P082            2192536
##  9 2102111799         2147570
## 10 C01P038            2054406
## # i 78 more rows
```

Normalisations

```r
  # On créer une matrice ge pour edgeR
comptages_larges <- comptages_long |> group_by(nom_gene, echantillon) |> summarise(total_lectures =
```

TPM

```r
tpm_long <- comptages_long |> group_by(echantillon) |> mutate(
  taux_expression = lectures_brutes / Longueur,
  TPM_transcrits_par_M = taux_expression / sum(taux_expression) * 1e6) |> ungroup ()
```

TMM (Trimmed Mean of M-values)

```
objet_tmm_edge <- DGEList(counts = as.matrix(comptages_larges))
objet_tmm_edge <- calcNormFactors(objet_tmm_edge, method ="TMM")

matrice_cpm_tmm <- cpm(objet_tmm_edge, normalized.lib.sizes = TRUE)
```

RLE (Relative Log expression)

```
objet_rle_edge <- DGEList(counts = as.matrix(comptages_larges))
objet_rle_edge <- calcNormFactors(objet_rle_edge, method ="RLE")

matrice_cpm_rle <- cpm(objet_rle_edge, normalized.lib.size = TRUE)
```

Création d'un tibble global des comptes normalisés

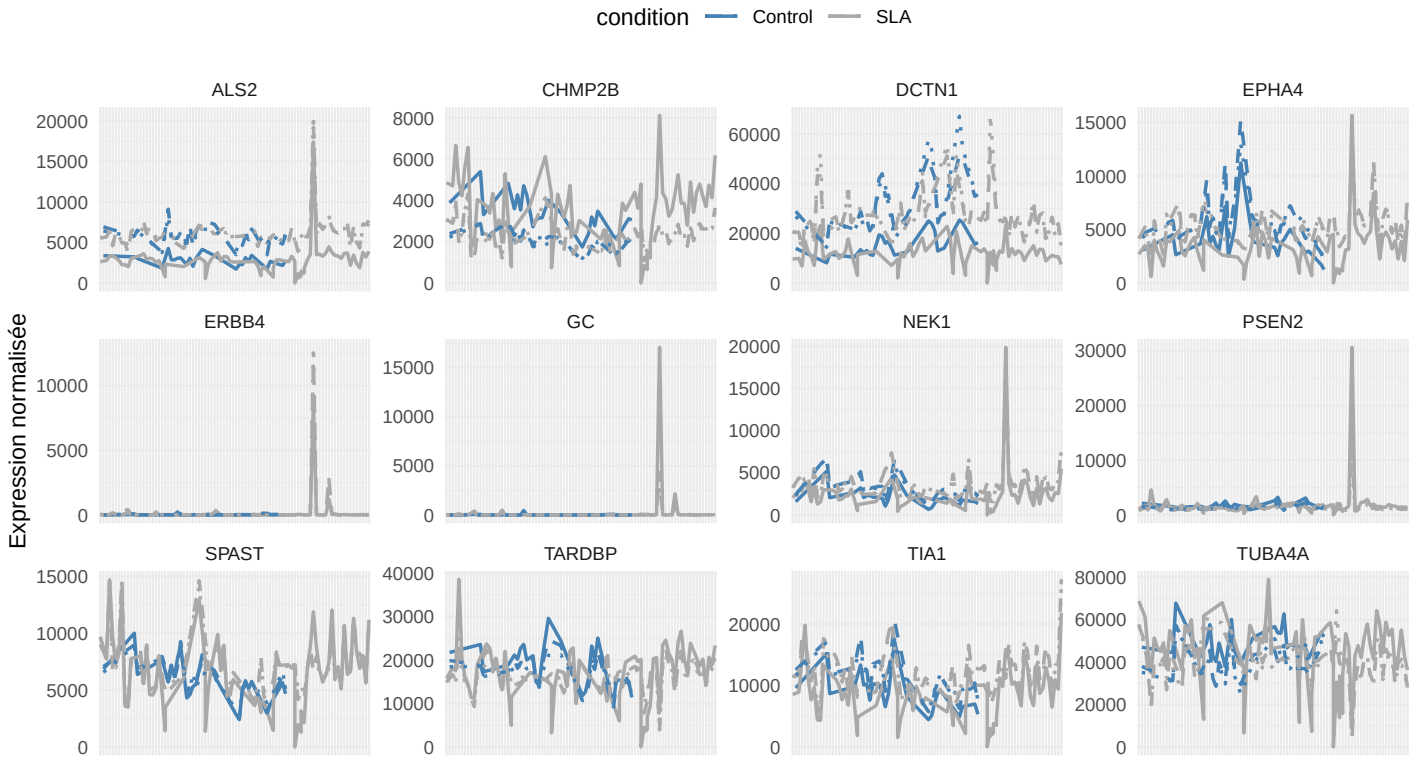
```
tpm_tb <- tpm_long |> select(echantillon, nom_gene, TPM_transcrits_par_M)

tmm_tb <- matrice_cpm_tmm |> as.data.frame() |> rownames_to_column("nom_gene") |> pivot_longer(-nom_

rle_tb <- matrice_cpm_rle |> as.data.frame() |> rownames_to_column("nom_gene") |> pivot_longer(-nom_
names_to = "echantillon", values_to = "RLE_cpm")

donnees_normalisees <- reduce(list(tpm_tb, tmm_tb, rle_tb), full_join, by = c("echantillon","nom_ger
  left_join (comptages_long |> distinct(echantillon,run,condition), by ="echantillon")
```

Visualisation des comptes normalisés

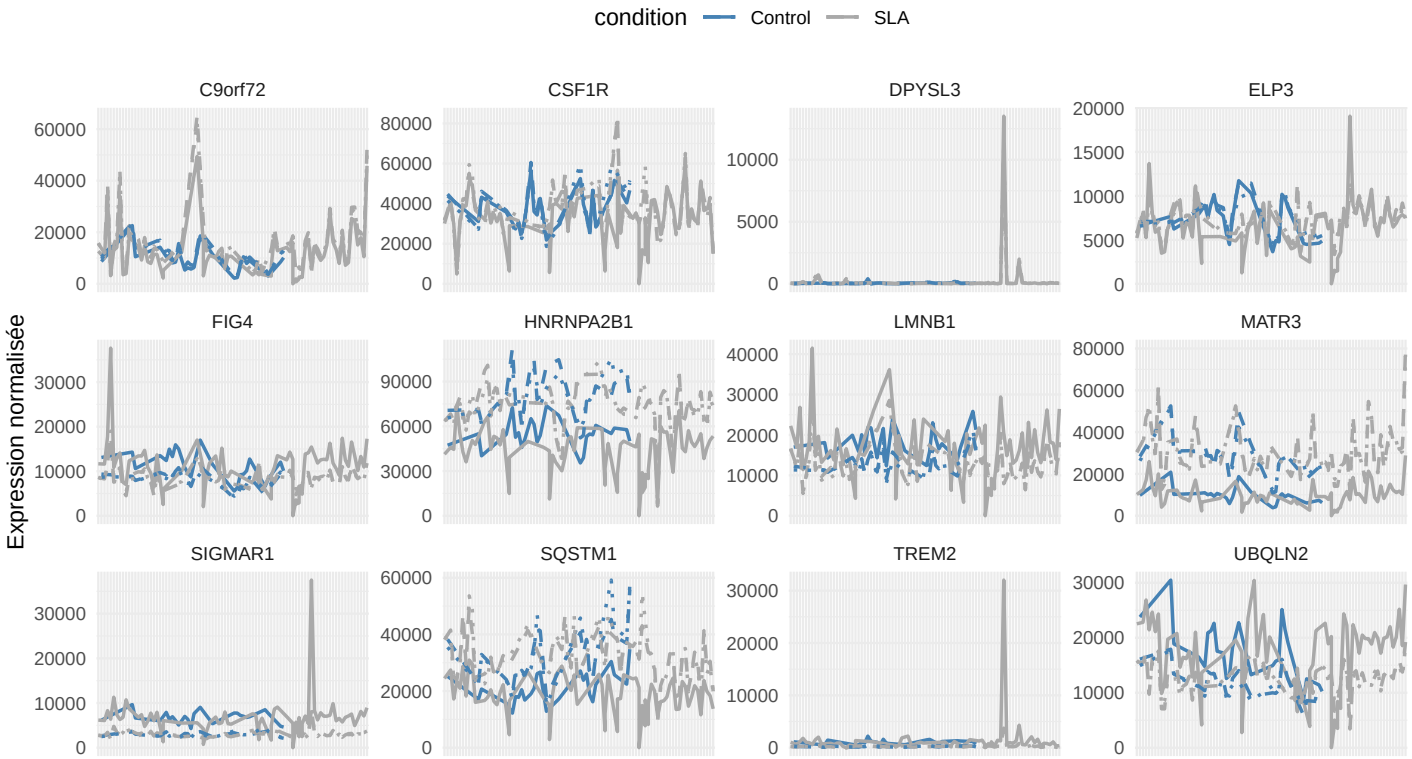
On regarde a premiere vue comment les comptes normalisés ce répartissent par gènes, avec une facet par gène pour les trois approches. L'idée étant surtout ici d'observer si il y'a des différences excessives entre les approches et comprendre dérrière à quoi elles peuvent êtres dues ...

```
liste_genes <- unique(donnees_normalisees$nom_gene)
groupes_genes <- split(liste_genes, ceiling(seq_along(liste_genes) / 12))

donnees_normalisees$condition <- factor(donnees_normalisees$condition, levels = c("Control", "SLA"))

for (indice_pan in seq_along(groupes_genes)) {
  sous_ensemble <- donnees_normalisees %>% filter(nom_gene %in% groupes_genes[[indice_pan]])

  p <- ggplot(sous_ensemble, aes(x = echantillon)) +
    geom_line(aes(y = TPM_transcrits_par_M, color = condition,
                  group = interaction(condition, "TPM")), linewidth = 0.7) +
    geom_line(aes(y = TMM_cpm, color = condition,
                  group = interaction(condition, "TMM")), linewidth = 0.7, linetype = "dashed") +
    geom_line(aes(y = RLE_cpm, color = condition,
                  group = interaction(condition, "RLE")), linewidth = 0.7, linetype = "dotted") +
    facet_wrap(~ nom_gene, scales = "free_y") +
    theme_minimal(base_size = 10) +
    theme(axis.text.x = element_blank(), legend.position = "top") +
```

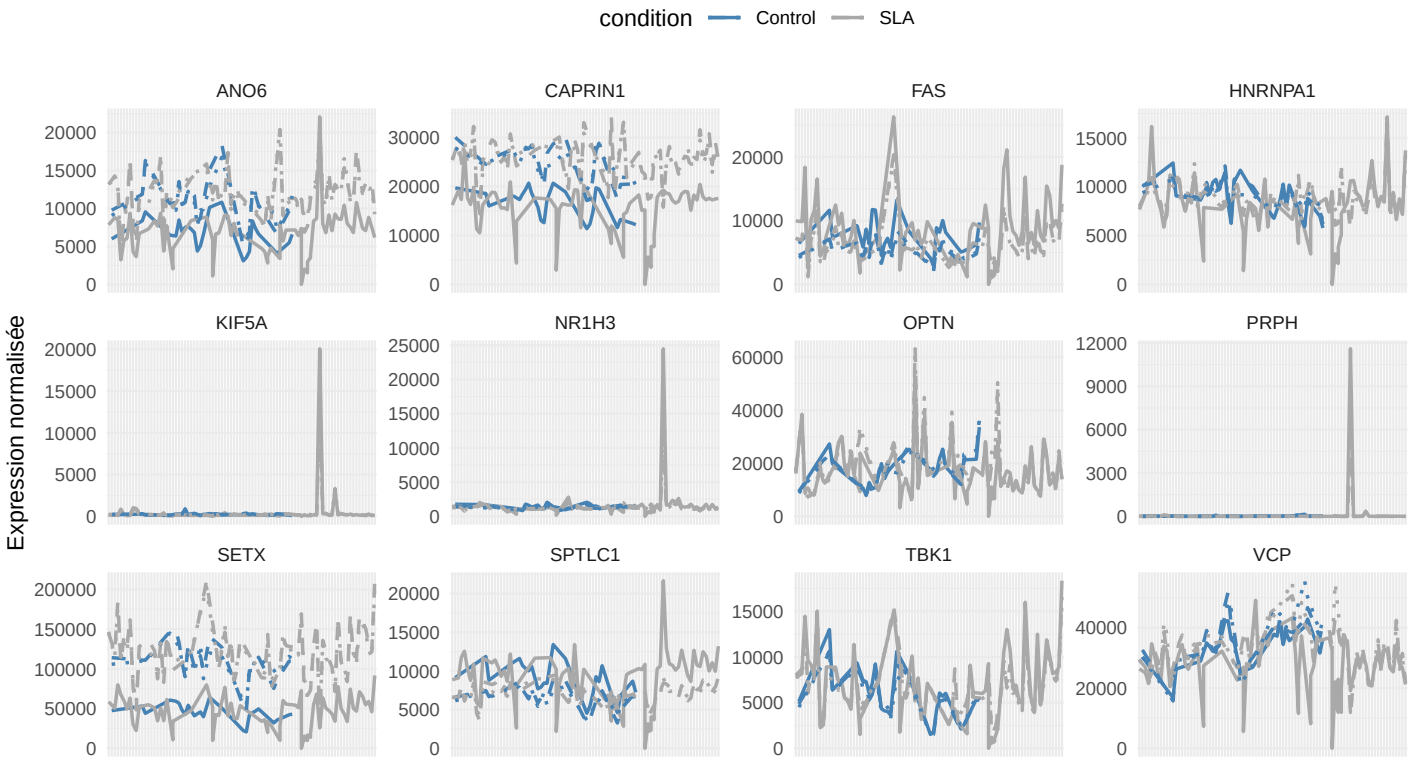
```
    scale_color_manual(values = c("Control" = "steelblue", "SLA" = "darkgray")) +
    labs(title = sprintf("Comparaison TPM / TMM / RLE par condition - Panel %d", indice_pan),
         y = "Expression normalisée", x = NULL)

  print(p)
}
```

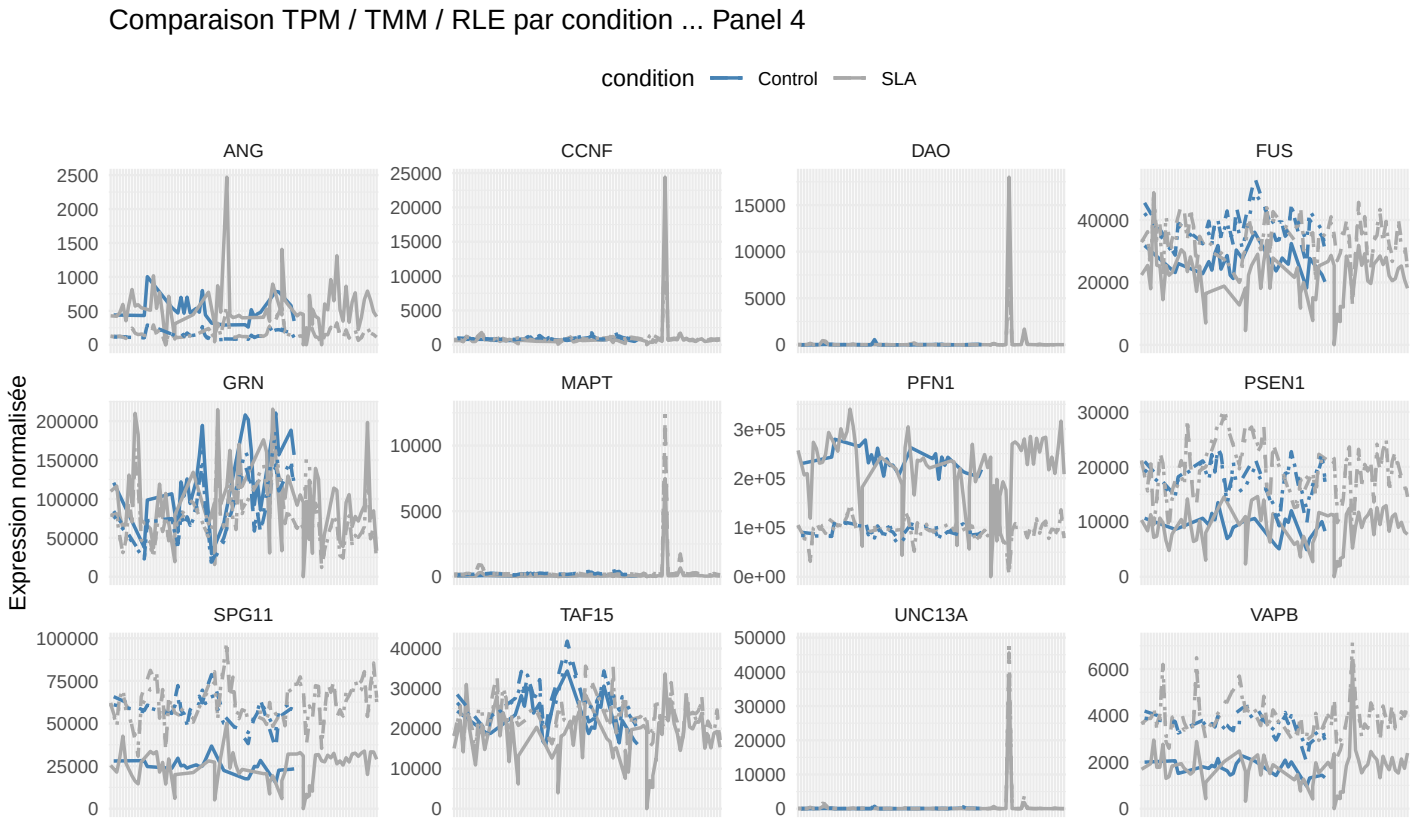


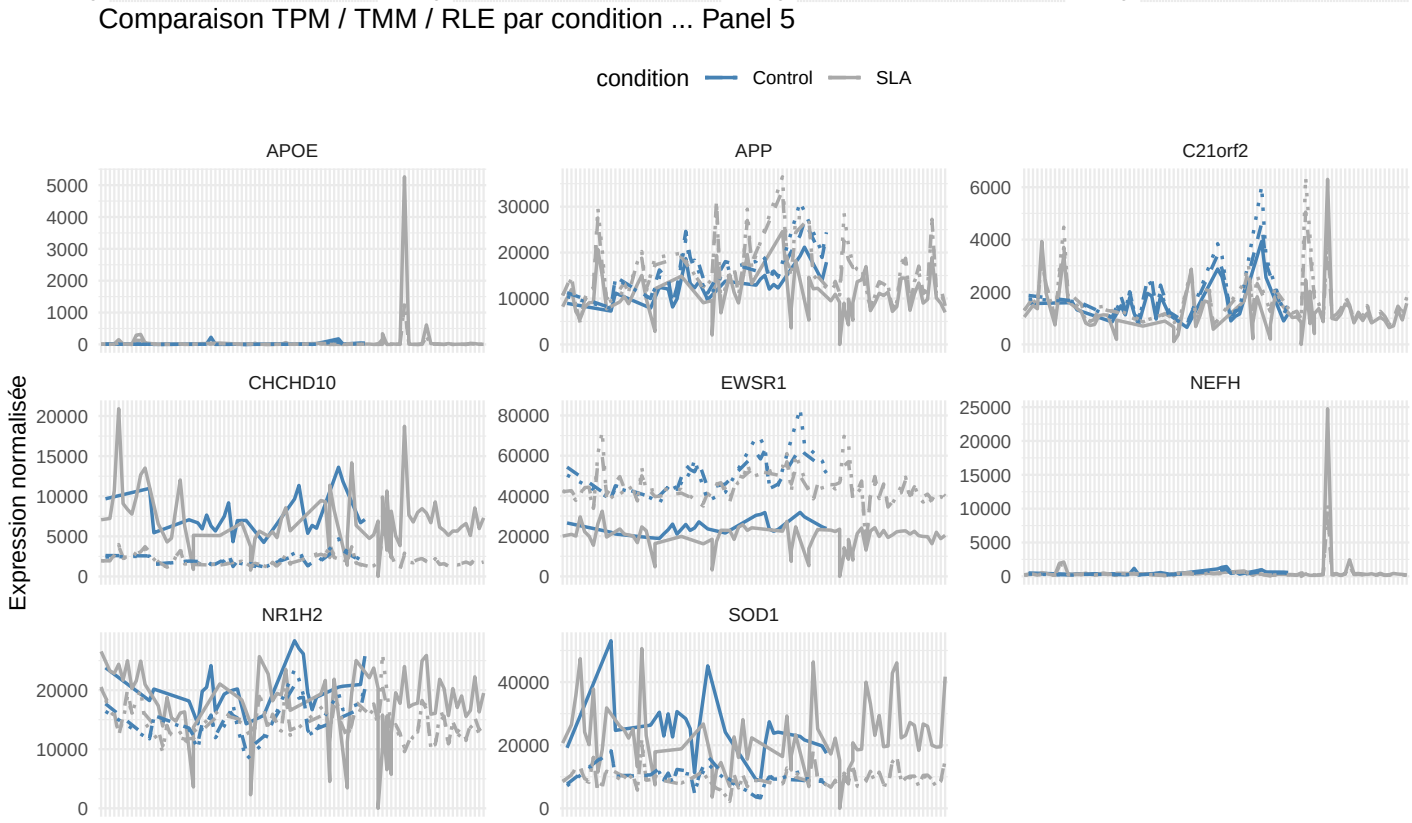
Comparaison TPM / TMM / RLE par condition ... Panel 2



Comparaison TPM / TMM / RLE par condition ... Panel 3




Comparaison TPM / TMM / RLE par condition ... Panel 4
condition
Control
SLA
ANG
CCNF
DAO
FUS
GRN
MAPT
PFN1
PSEN1
SPG11
TAF15
UNC13A
VAPB
Expression normalisée


Comparaison TPM / TMM / RLE par condition ... Panel 5
condition
Control
SLA
APOE
APP
C21orf2
CHCHD10
EWSR1
NEFH
NR1H2
SOD1
Expression normalisée

Références

Robinson, M.D., & Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. Genome Biology.

Li, B., & Dewey, C.N. (2011). RSEM: accurate transcript quantification from RNA-Seq data. BMC Bioinformatics.

Risso, D., etal. (2014). Normalization of RNA-seq data using factor analysis of control genes or samples. Nature Biotechnology.