Pertussis Challenge 3.1_CCL+KAT+STAT

Runqi Zhang

2024-11-21

```r
# Load libraries and suppress unnecessary output
suppressPackageStartupMessages({
  library(dplyr)
  library(tidyr)
  library(readr)
  library(agua)          # H2O AutoML
  library(edgeR)         # For TMM normalization
  library(GSVA)          # Gene set variation analysis
  library(biomaRt)       # Gene ID mapping
  library(glmnet)        # Regression
  library(knitr)         # Output formatting
  library(tibble)        # For handling row names
  library(impute)
})
```

```
## Warning: package 'tidyr' was built under R version 4.3.3

## Warning: package 'readr' was built under R version 4.3.3

## Warning: package 'agua' was built under R version 4.3.3

## Warning: package 'parsnip' was built under R version 4.3.3

## Warning: package 'GSVA' was built under R version 4.3.3

## Warning: package 'biomaRt' was built under R version 4.3.2

## Warning: package 'glmnet' was built under R version 4.3.3

## Warning: package 'knitr' was built under R version 4.3.3
```

```r
# Define working directory and initialize H2O
workDir <- "C:/Users/zhang/Desktop/cmi-pb-3rd-final/Runqi/CMI-PB"
options(readr.show_col_types = FALSE)
agua::h2o_start()
```

```
## Warning: JAVA not found, H2O may take minutes trying to connect.

## Warning in h2o.clusterInfo():
## Your H2O cluster version is (11 months and 1 day) old. There may be a newer version available.
## Please download and install the latest version from: https://h2o-release.s3.amazonaws.com/h2o/latest
```

```r
# Function to read input data for a given year
read_data <- function(year, type = "LD") {
  list(
    pts = read_tsv(file.path(workDir, paste0("data/", year, type, "_subject.tsv"))),
    sample = read_tsv(file.path(workDir, paste0("data/", year, type, "_specimen.tsv"))),
    ge = read_tsv(file.path(workDir, paste0("data/", year, type, "_pbmc_gene_expression.tsv")))
  )
}

# Load datasets for 2020-2023
data2020 <- read_data("2020")
data2021 <- read_data("2021")
data2022 <- read_data("2022")
data2023 <- read_data("2023", type = "BD")
```

```r
# Function to prepare outcome data for CCL3 on day 3
prepare_outcome_day3 <- function(ge_data, sample_data, pts_data, gene_id, day = 3) {
  ge_data %>%
    filter(versioned_ensembl_gene_id == gene_id) %>%
    inner_join(sample_data, by = "specimen_id") %>%
    inner_join(pts_data, by = "subject_id") %>%
    filter(planned_day_relative_to_boost == day) %>%
    dplyr::select(subject_id, tpm) %>%
    mutate(tpm_day3 = scale(tpm)) # Scale the TPM values for modeling
}

# Prepare outcome data for all years
yDF <- prepare_outcome_day3(data2020$ge, data2020$sample, data2020$pts, "ENSG00000277632.1")
y2DF <- prepare_outcome_day3(data2021$ge, data2021$sample, data2021$pts, "ENSG00000277632.1")
y3DF <- prepare_outcome_day3(data2022$ge, data2022$sample, data2022$pts, "ENSG00000277632.1")
```

#top 20 CCL paralogs #CCL3 ENSG00000277632.1 #CCL3L3 ENSG00000276085.1 #CCL4 ENSG00000275302.1 #CCL4L2 ENSG00000276070.4 #CCL5 ENSG00000271503.5 #CCL14 ENSG00000276409.4 #CCL15 ENSG00000275718.1 #CCL18 ENSG00000275385.1 #CCL22 ENSG00000102962.4 #CCL23 ENSG00000274736.4 #CCL26 ENSG00000006606.8 #CCL24 ENSG00000106178.6 #CCL16 ENSG00000275152.4 #CCL1 ENSG00000108702.1 #CCL17 ENSG00000102970.10 #CCL25 ENSG00000131142.14 #CCL7 ENSG00000108688.11 #CCL19 ENSG00000172724.11 #CCL8 ENSG00000108700.4 #XCL1 ENSG00000143184.4 #XCL2 ENSG00000143185.3

#KAT -> STAT1 -> CCL3 pathway #KAT7 ENSG00000136504.11 #PHF10 ENSG00000130024.14 #RSF1 ENSG00000048649.14 #KAT6B ENSG00000156650.12 #DPF1 ENSG00000011332.19 #KAT5 ENSG00000172977.12 #KAT8 ENSG00000103510.19 #DPF3 ENSG00000205683.11 #KAT6A ENSG00000083168.9 #DPF2 ENSG00000133884.9

#STAT1 ENSG00000115415.18 #STAT2 ENSG00000170581.13 #STAT3 ENSG00000168610.14 #STAT4 ENSG00000138378.17 #STAT5A ENSG00000126561.16 #STAT6 ENSG00000166888.11 #STAT5B ENSG00000173757.9

```r
# Target genes
target_genes <- c(
  "ENSG00000277632.1", # CCL3
  "ENSG00000276085.1", # CCL3L3
  "ENSG00000275302.1", # CCL4
  "ENSG00000276070.4", # CCL4L2
  "ENSG00000271503.5", # CCL5
  "ENSG00000276409.4", # CCL14
  "ENSG00000275718.1", # CCL15
  "ENSG00000275385.1", # CCL18
  "ENSG00000102962.4", # CCL22
  "ENSG00000274736.4", # CCL23
  "ENSG00000006606.8", #CCL26
  "ENSG00000106178.6", #CCL24
  "ENSG00000275152.4", #CCL16
  "ENSG00000108702.1", #CCL1
  "ENSG00000102970.10", #CCL17
  "ENSG00000131142.14", #CCL25
  "ENSG00000108688.11", #CCL7
  "ENSG00000172724.11", #CCL19
  "ENSG00000108700.4", #CCL8
  "ENSG00000143184.4", #XCL1
  "ENSG00000143185.3", #XCL2
  "ENSG00000136504.11", #KAT7
  "ENSG00000130024.14", #PHF10
  "ENSG00000048649.14", #RSF1
  "ENSG00000156650.12", #KAT6B
  "ENSG00000011332.19", #DPF1
  "ENSG00000172977.12", #KAT5
  "ENSG00000103510.19", #KAT8
  "ENSG00000205683.11", #DPF3
  "ENSG00000083168.9", #KAT6A
  "ENSG00000133884.9", #DPF2
  "ENSG00000115415.18", #STAT1
  "ENSG00000170581.13", #STAT2
  "ENSG00000168610.14", #STAT3
  "ENSG00000138378.17", #STAT4
  "ENSG00000126561.16", #STAT5A
  "ENSG00000166888.11", #STAT6
  "ENSG00000173757.9" #STAT5B
)

# Function to extract day 0 TPM values
extract_day0_tpm <- function(ge_data, sample_data, pts_data, target_genes) {
  ge_data %>%
    filter(versioned_ensembl_gene_id %in% target_genes) %>%
    inner_join(sample_data, by = "specimen_id") %>%
    inner_join(pts_data, by = "subject_id") %>%
    filter(planned_day_relative_to_boost == 0) %>%
    dplyr::select(subject_id, versioned_ensembl_gene_id, tpm) %>%
    pivot_wider(
      names_from = versioned_ensembl_gene_id,
      values_from = tpm,
      names_prefix = "tpm_"
    ) %>%
    replace(is.na(.), 0) %>% # Replace NA with 0
    column_to_rownames("subject_id") # Use subject_id as rownames
}
```

```r
# Extract day 0 TPM values for each dataset
xDF <- extract_day0_tpm(data2020$ge, data2020$sample, data2020$pts, target_genes)
x2DF <- extract_day0_tpm(data2021$ge, data2021$sample, data2021$pts, target_genes)
x3DF <- extract_day0_tpm(data2022$ge, data2022$sample, data2022$pts, target_genes)
x4DF <- extract_day0_tpm(data2023$ge, data2023$sample, data2023$pts, target_genes)

# Ensure consistent columns across datasets
common_cols <- Reduce(intersect, list(colnames(xDF), colnames(x2DF), colnames(x3DF), colnames(x4DF)))
xDF <- xDF[, common_cols]
x2DF <- x2DF[, common_cols]
x3DF <- x3DF[, common_cols]
x4DF <- x4DF[, common_cols]
```

```r
# Train model with H2O AutoML
# Combine training data and outcome variable for TPM of CCL3 on day 3
trainDF <- rbind(xDF, x2DF, x3DF) %>%
  as.data.frame() %>%
  mutate(tpm_day3 = c(yDF$tpm_day3, y2DF$tpm_day3, y3DF$tpm_day3))

# Impute missing values if needed
train_matrix <- as.matrix(trainDF) # Convert to matrix
imputed_matrix <- impute.knn(train_matrix)$data
trainDF <- as.data.frame(imputed_matrix) # Convert back to dataframe

# Train the model
set.seed(3)
auto_fit <- auto_ml() %>%
  set_engine("h2o", max_runtime_secs = 5) %>%
  set_mode("regression") %>%
  fit(tpm_day3 ~ ., data = trainDF)
```

```r
# Validate model on training data
train_predictions <- predict(auto_fit, new_data = trainDF)$.pred

# Calculate correlations
pearson_cor <- cor(train_predictions, trainDF$tpm_day3, method = "pearson")
spearman_cor <- cor(train_predictions, trainDF$tpm_day3, method = "spearman")

# Display correlation results
cat("Pearson Correlation: ", pearson_cor, "/n")
```

```
## Pearson Correlation:  0.7726918 /n
```

```r
cat("Spearman Correlation: ", spearman_cor, "/n")
```

```
## Spearman Correlation:  0.6847206 /n
```

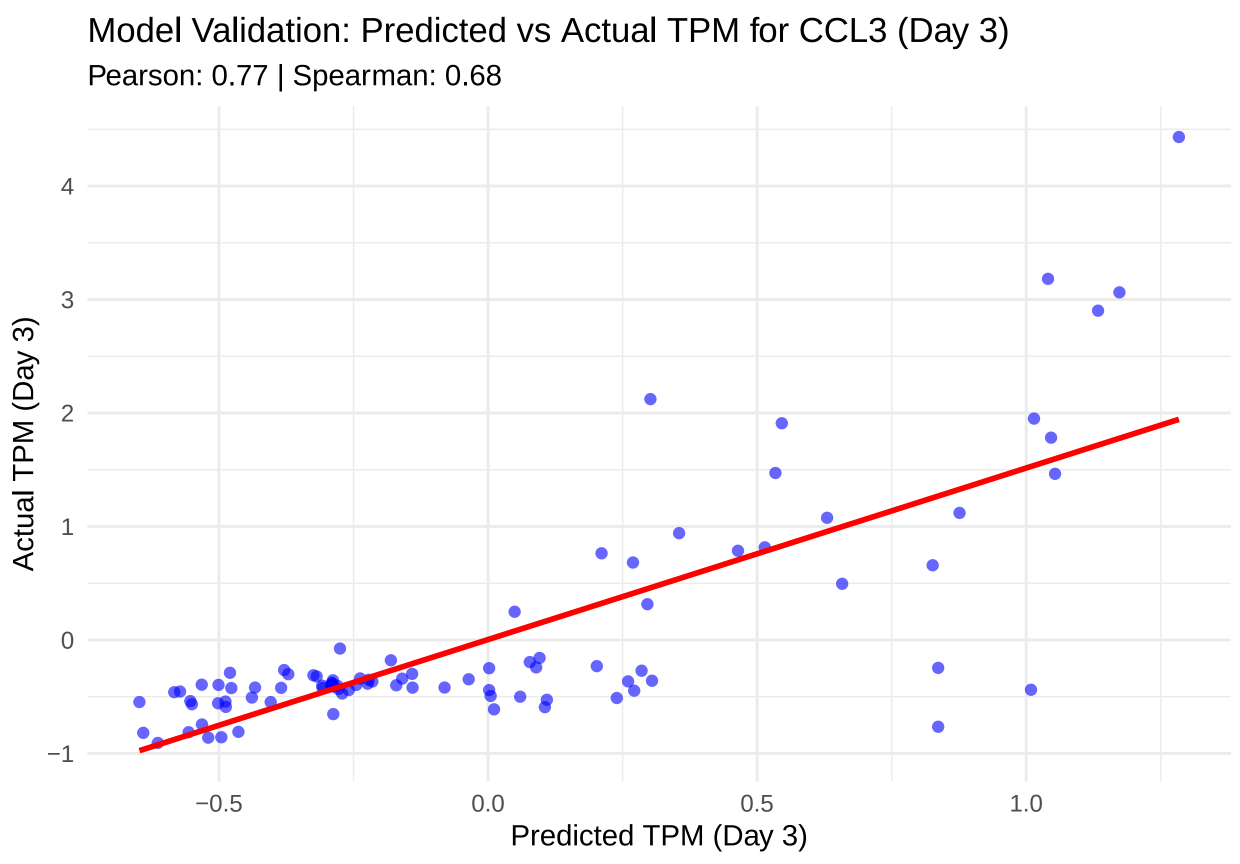
```r
# Plot validation results
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```r
correlation_plot <- ggplot(data = data.frame(
  Predicted = train_predictions,
  Actual = trainDF$tpm_day3
), aes(x = Predicted, y = Actual)) +
  geom_point(alpha = 0.6, color = "blue") +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(
    title = "Model Validation: Predicted vs Actual TPM for CCL3 (Day 3)",
    subtitle = paste0("Pearson: ", round(pearson_cor, 2),
                      " | Spearman: ", round(spearman_cor, 2)),
    x = "Predicted TPM (Day 3)",
    y = "Actual TPM (Day 3)"
  ) +
  theme_minimal()

# Display the plot
print(correlation_plot)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```r
# Train model with H2O AutoML
# Combine training data and outcome variable for TPM of CCL3 on day 3
trainDF <- rbind(xDF, x2DF, x3DF) %>%
  as.data.frame() %>%
  mutate(tpm_day3 = c(yDF$tpm_day3, y2DF$tpm_day3, y3DF$tpm_day3))

trainDF$tpm_day3 <- (log1p(trainDF$tpm_day3))

# Impute missing values if needed
train_matrix <- as.matrix(trainDF) # Convert to matrix
imputed_matrix <- impute.knn(train_matrix)$data
trainDF <- as.data.frame(imputed_matrix) # Convert back to dataframe

# Train the model
set.seed(3)
auto_fit <- auto_ml() %>%
  set_engine("h2o", max_runtime_secs = 5) %>%
  set_mode("regression") %>%
  fit(tpm_day3 ~ ., data = trainDF)
# Validate model on training data
train_predictions <- predict(auto_fit, new_data = trainDF)$.pred

# Calculate correlations
pearson_cor <- cor(train_predictions, trainDF$tpm_day3, method = "pearson")
spearman_cor <- cor(train_predictions, trainDF$tpm_day3, method = "spearman")

# Display correlation results
cat("Pearson Correlation: ", pearson_cor, "/n")
```

```
## Pearson Correlation:  0.9994129 /n
```

```r
cat("Spearman Correlation: ", spearman_cor, "/n")
```

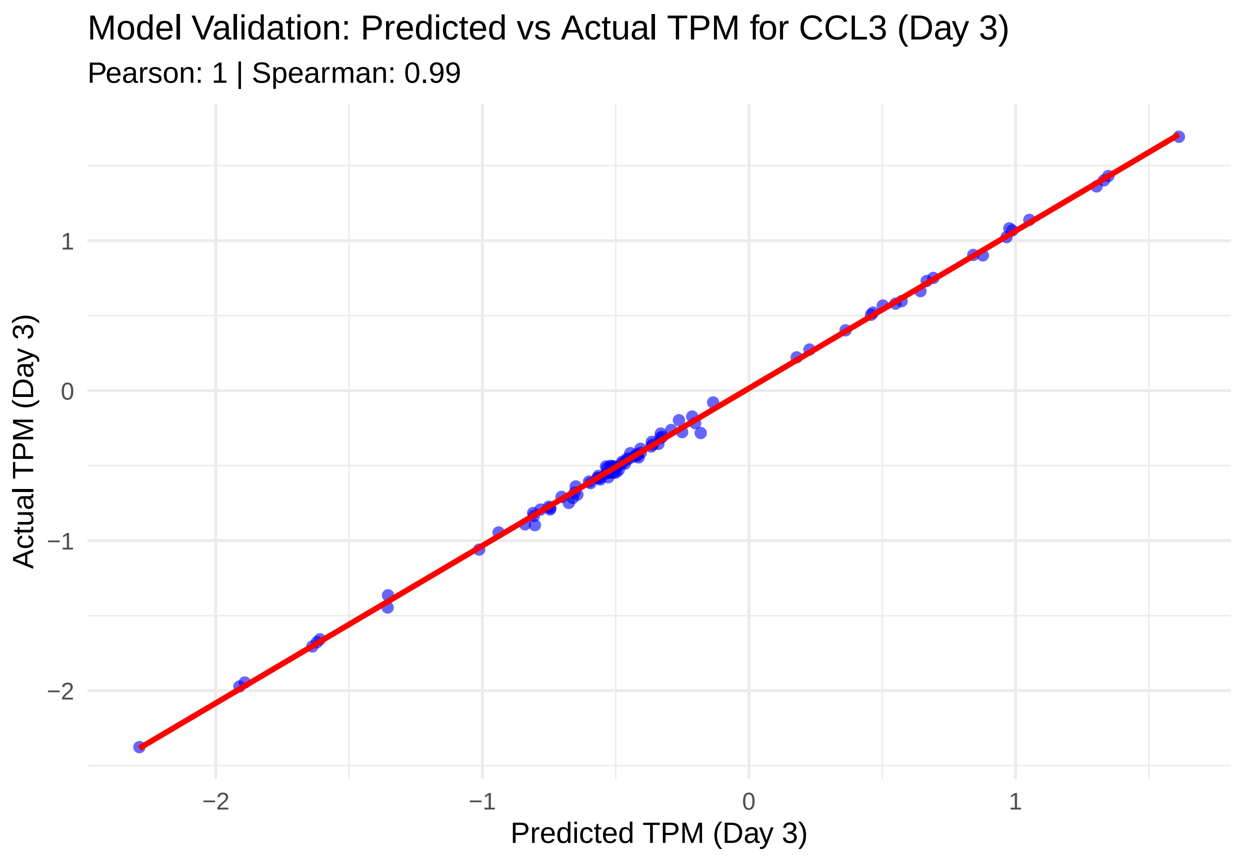
```
## Spearman Correlation:  0.9945689 /n
```

```r
# Plot validation results
library(ggplot2)

correlation_plot <- ggplot(data = data.frame(
  Predicted = train_predictions,
  Actual = trainDF$tpm_day3
), aes(x = Predicted, y = Actual)) +
  geom_point(alpha = 0.6, color = "blue") +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(
    title = "Model Validation: Predicted vs Actual TPM for CCL3 (Day 3)",
    subtitle = paste0("Pearson: ", round(pearson_cor, 2),
                      " | Spearman: ", round(spearman_cor, 2)),
    x = "Predicted TPM (Day 3)",
    y = "Actual TPM (Day 3)"
  ) +
  theme_minimal()

# Display the plot
print(correlation_plot)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# Predict and rank
yhat <- predict(auto_fit, new_data = x4DF)$.pred
rhat <- rank(-1 * yhat, ties.method = "first")
print(cbind(rownames(x4DF), rhat))
```

```
##             rhat
##  [1,] "129" "19"
##  [2,] "123" "45"
##  [3,] "136" "50"
##  [4,] "130" "11"
##  [5,] "121" "44"
##  [6,] "122" "48"
##  [7,] "127" "47"
##  [8,] "124" "36"
##  [9,] "125" "4"
## [10,] "126" "46"
## [11,] "128" "52"
## [12,] "131" "53"
## [13,] "132" "41"
## [14,] "133" "51"
## [15,] "134" "28"
## [16,] "137" "49"
## [17,] "138" "43"
## [18,] "140" "14"
## [19,] "144" "23"
## [20,] "139" "18"
## [21,] "141" "39"
## [22,] "135" "32"
## [23,] "149" "22"
## [24,] "143" "24"
## [25,] "150" "35"
## [26,] "142" "40"
## [27,] "145" "10"
## [28,] "146" "29"
## [29,] "147" "31"
## [30,] "153" "37"
## [31,] "154" "3"
## [32,] "170" "21"
## [33,] "172" "20"
## [34,] "162" "42"
## [35,] "119" "34"
## [36,] "160" "38"
## [37,] "148" "33"
## [38,] "151" "17"
## [39,] "152" "13"
## [40,] "155" "27"
## [41,] "156" "2"
## [42,] "157" "1"
## [43,] "158" "12"
## [44,] "159" "8"
## [45,] "161" "26"
## [46,] "165" "25"
## [47,] "163" "5"
## [48,] "164" "15"
## [49,] "166" "9"
## [50,] "167" "6"
## [51,] "168" "7"
## [52,] "169" "16"
## [53,] "171" "30"
```

```
# Update submission file with rankings
submission_file <- file.path(workDir, "3rdChallengeSubmissionTemplate_revised.tsv")
data <- read_tsv(submission_file)

ranking_df <- data.frame(
  SubjectID = as.numeric(rownames(x4DF)),
  "3.1) CCL3-D3-Rank" = rhat,
  check.names = FALSE
)

data <- data %>%
  mutate(
    `3.1) CCL3-D3-Rank` = ifelse(
      SubjectID %in% ranking_df$SubjectID,
      ranking_df$`3.1) CCL3-D3-Rank`[match(SubjectID, ranking_df$SubjectID)],
      `3.1) CCL3-D3-Rank`
    )
```

```r
  )

write_tsv(data, submission_file)
```

```r
# End H2O session and print session info
agua::h2o_end()
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 11 x64 (build 22631)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/Los_Angeles
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggplot2_3.5.1  impute_1.76.0  tibble_3.2.1   knitr_1.49     glmnet_4.1-8
##  [6] Matrix_1.5-4.1 biomaRt_2.58.2 GSVA_1.50.5    edgeR_3.42.4   limma_3.58.1
## [11] agua_0.1.4     parsnip_1.2.1  readr_2.1.5    tidyr_1.3.1    dplyr_1.1.3
##
## loaded via a namespace (and not attached):
##   [1] jsonlite_1.8.9              shape_1.4.6.1
##   [3] rstudioapi_0.17.1           magrittr_2.0.3
##   [5] farver_2.1.2                rmarkdown_2.29
##   [7] zlibbioc_1.46.0             vctrs_0.6.3
##   [9] memoise_2.0.1               DelayedMatrixStats_1.24.0
##  [11] RCurl_1.98-1.16             progress_1.2.3
##  [13] htmltools_0.5.8.1           S4Arrays_1.2.1
##  [15] dials_1.3.0                 curl_6.0.1
##  [17] Rhdf5lib_1.24.2             SparseArray_1.2.4
##  [19] rhdf5_2.46.1                parallelly_1.39.0
##  [21] lubridate_1.9.3             cachem_1.1.0
##  [23] lifecycle_1.0.4             iterators_1.0.14
##  [25] pkgconfig_2.0.3             rsvd_1.0.5
##  [27] R6_2.5.1                    fastmap_1.2.0
##  [29] GenomeInfoDbData_1.2.11     MatrixGenerics_1.14.0
##  [31] future_1.34.0               tune_1.2.1
##  [33] digest_0.6.37               colorspace_2.1-0
##  [35] furrr_0.3.1                 AnnotationDbi_1.64.1
##  [37] S4Vectors_0.38.2            irlba_2.3.5.1
##  [39] GenomicRanges_1.54.1        RSQLite_2.3.7
##  [41] beachmat_2.18.1             labeling_0.4.3
##  [43] filelock_1.0.3              fansi_1.0.4
##  [45] yardstick_1.3.1             timechange_0.3.0
##  [47] mgcv_1.8-42                 httr_1.4.7
##  [49] abind_1.4-8                 compiler_4.3.1
##  [51] bit64_4.5.2                 withr_3.0.2
##  [53] BiocParallel_1.34.2         DBI_1.2.3
##  [55] HDF5Array_1.30.1            MASS_7.3-60
##  [57] lava_1.8.0                  rappdirs_0.3.3
##  [59] DelayedArray_0.28.0         tools_4.3.1
##  [61] future.apply_1.11.3         nnet_7.3-19
##  [63] glue_1.6.2                  nlme_3.1-162
##  [65] h2o_3.44.0.3                rhdf5filters_1.14.1
##  [67] grid_4.3.1                  generics_0.1.3
##  [69] recipes_1.1.0               gtable_0.3.6
##  [71] tzdb_0.4.0                  class_7.3-22
##  [73] data.table_1.16.2           hms_1.1.3
##  [75] rsample_1.2.1               xml2_1.3.6
##  [77] BiocSingular_1.18.0         ScaledMatrix_1.10.0
##  [79] utf8_1.2.3                  XVector_0.40.0
##  [81] BiocGenerics_0.48.1         stringr_1.5.1
##  [83] foreach_1.5.2               pillar_1.9.0
##  [85] vroom_1.6.5                 splines_4.3.1
##  [87] lhs_1.2.0                   BiocFileCache_2.10.2
##  [89] lattice_0.21-8              survival_3.5-5
##  [91] bit_4.5.0                   annotate_1.80.0
##  [93] tidyselect_1.2.1            SingleCellExperiment_1.24.0
##  [95] locfit_1.5-9.10             Biostrings_2.70.3
##  [97] IRanges_2.34.1              SummarizedExperiment_1.32.0
##  [99] stats4_4.3.1                xfun_0.48
## [101] Biobase_2.62.0              statmod_1.5.0
## [103] hardhat_1.4.0               timeDate_4041.110
## [105] matrixStats_1.4.1           stringi_1.8.4
## [107] DiceDesign_1.10             yaml_2.3.10
## [109] workflows_1.1.4             evaluate_1.0.1
## [111] codetools_0.2-19            graph_1.80.0
## [113] cli_3.6.1                   rpart_4.1.19
## [115] xtable_1.8-4                munsell_0.5.1
## [117] Rcpp_1.0.11                 GenomeInfoDb_1.38.8
## [119] globals_0.16.3              dbplyr_2.5.0
## [121] png_0.1-8                   XML_3.99-0.17
## [123] parallel_4.3.1              gower_1.0.1
## [125] blob_1.2.4                  prettyunits_1.2.0
## [127] sparseMatrixStats_1.14.0    bitops_1.0-9
## [129] GPfit_1.0-8                 listenv_0.9.1
## [131] GSEABase_1.64.0             ipred_0.9-15
## [133] scales_1.3.0                prodlim_2024.06.25
## [135] purrr_1.0.2                 crayon_1.5.3
## [137] rlang_1.1.1                 KEGGREST_1.42.0
```