Day 2 – Exercise Solutions

Seminar reference solutions

Use these solutions after the workshop to debrief or double-check the exercises (bonus included).

Solution – Explore Data

Use this key after attempting `scripts/individual_notebooks/01_explore_data_exercises.Rmd`. The code chunks mirror the TODOs but include one possible solution for each task.

1. Load and preview the dataset

```
input_path <- file.path('..', 'data', 'dataset2_subset_long.csv')
long_df <- read.csv(input_path, stringsAsFactors = FALSE, check.names = FALSE)
cat('Rows:', nrow(long_df), '\\nColumns:', ncol(long_df), '\\n')
```

```
## Rows: 3072 \nColumns: 7 \n
```

```
head(long_df)
```

```
##                         Genome      snp_id Position    value mouse_id day
## 1 Akkermansia_muciniphila_YL44  239840-C-G   239840 0.000000     1683   0
## 2 Akkermansia_muciniphila_YL44  241793-A-G   241793 0.049587     1683   0
## 3 Akkermansia_muciniphila_YL44  355328-A-T   355328 0.138182     1683   0
## 4 Akkermansia_muciniphila_YL44  356291-C-A   356291 0.000000     1683   0
## 5 Akkermansia_muciniphila_YL44 2351445-C-T  2351445 0.000000     1683   0
## 6   Bacteroides_caecimuris_I48 1601848-T-C  1601848 0.041609     1683   0
##   treatment_group
## 1         Control
## 2         Control
## 3         Control
## 4         Control
## 5         Control
## 6         Control
```

2. Enumerate genomes and SNPs

```
unique(long_df$Genome)
```

```
## [1] "Akkermansia_muciniphila_YL44" "Bacteroides_caecimuris_I48"
## [3] "Turicimonas_muris_YL45"
```

```
table(long_df$Genome)
```

```
##
## Akkermansia_muciniphila_YL44   Bacteroides_caecimuris_I48
```

```
##                          320                          1216
##       Turicimonas_muris_YL45
##                         1536
```

```
tapply(long_df$snp_id, long_df$Genome, function(x) length(unique(x)))
```

```
## Akkermansia_muciniphila_YL44   Bacteroides_caecimuris_I48
##                            5                           19
##       Turicimonas_muris_YL45
##                           24
```

3. Summaries by genome

```
sample_values <- long_df$value[1:10]
summary(sample_values)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.03426 0.04759 0.13818
```

```
aggregate(value ~ Genome, data = long_df, function(x) summary(x))
```

```
##                         Genome value.Min. value.1st Qu. value.Median value.Mean
## 1 Akkermansia_muciniphila_YL44  0.0000000     0.0000000    0.0523090  0.1741893
## 2   Bacteroides_caecimuris_I48  0.0000000     0.0000000    0.0338305  0.1485411
## 3       Turicimonas_muris_YL45  0.0000000     0.0000000    0.2649125  0.4312851
##   value.3rd Qu. value.Max.
## 1     0.2714463  0.8712450
## 2     0.2679232  0.9912180
## 3     0.9090910  1.0000000
```

4. Focus on Turicimonas

```
turicimonas_df <- long_df[long_df$Genome == 'Turicimonas_muris_YL45', ]
cat('Rows for Turicimonas:', nrow(turicimonas_df), '\\n')
```

```
## Rows for Turicimonas: 1536 \n
```

```
head(turicimonas_df)
```

```
##                    Genome      snp_id Position    value mouse_id day
## 25 Turicimonas_muris_YL45  362534-G-A   362534 1.000000     1683   0
## 26 Turicimonas_muris_YL45 1063346-G-A  1063346 0.741935     1683   0
## 27 Turicimonas_muris_YL45 1135479-G-A  1135479 0.000000     1683   0
## 28 Turicimonas_muris_YL45 1364256-G-A  1364256 0.406250     1683   0
## 29 Turicimonas_muris_YL45 1376518-G-A  1376518 0.000000     1683   0
## 30 Turicimonas_muris_YL45 1392383-G-A  1392383 0.461538     1683   0
##    treatment_group
## 25         Control
## 26         Control
## 27         Control
## 28         Control
## 29         Control
## 30         Control
```

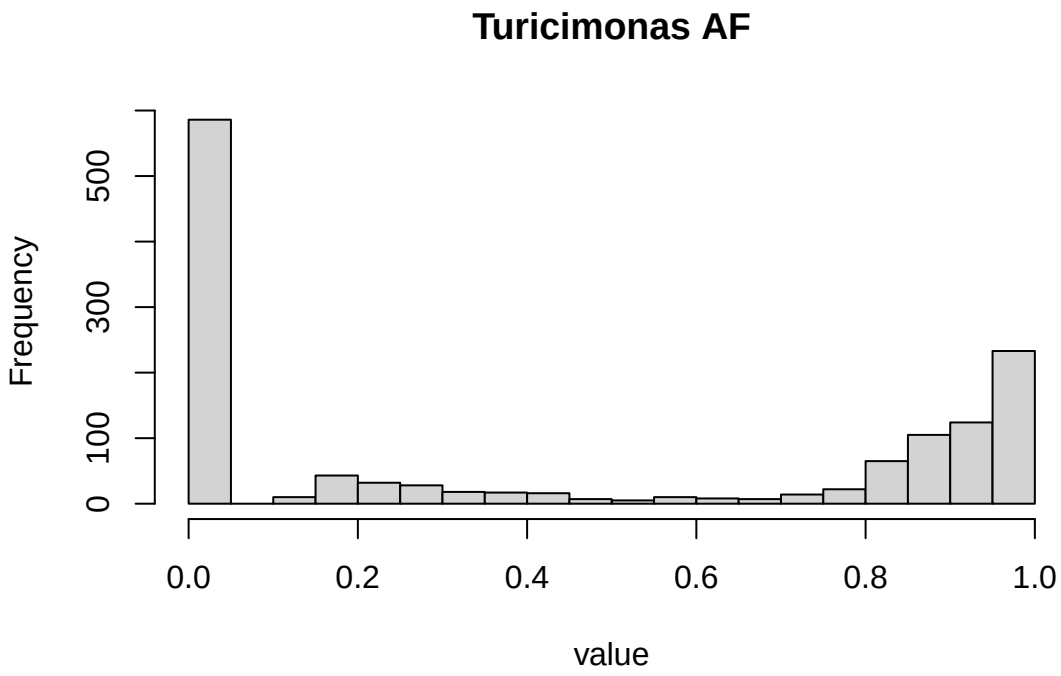
```
table(turicimonas_df$mouse_id, turicimonas_df$day)
```

```
##
```

```
##          0  4  9 14 18 23 30 37 44 49 53 58 63 67 72 79
##   1683 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
##   1688 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
##   1692 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
##   1699 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
```

5. Allele-frequency focus (Turicimonas vs all genomes)

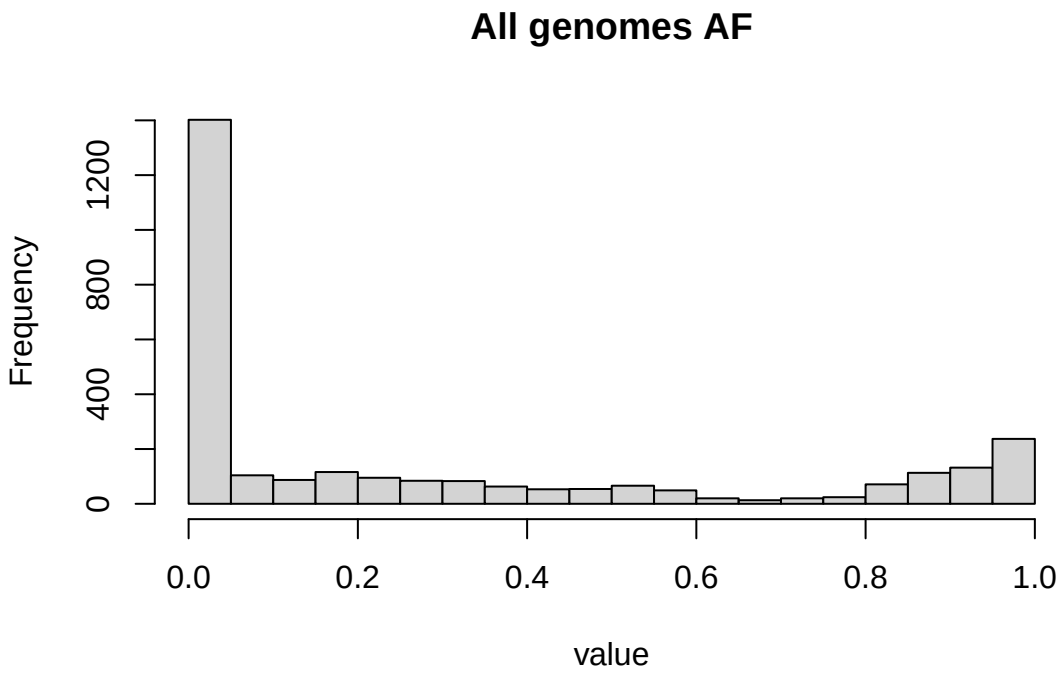
```
turicimonas_values <- long_df$value[long_df$Genome == 'Turicimonas_muris_YL45']
hist(turicimonas_values, breaks = 20, main = 'Turicimonas AF', xlab = 'value')
```




```
summary(turicimonas_values)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0000  0.0000  0.2649  0.4313  0.9091  1.0000     186
```

```
hist(long_df$value, breaks = 30, main = 'All genomes AF', xlab = 'value')
```




```
summary(long_df$value)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 0.00000 0.00000 0.06438 0.28365 0.51995 1.00000     186
```

6. Missing values

```
na_total <- sum(is.na(long_df$value))
na_total
```

```
## [1] 186
```

```
na_by_mouse_day <- with(long_df, tapply(value, list(mouse_id, day), function(x) sum(is.na(x))))
na_by_mouse_day
```

```
##       0  4  9 14 18 23 30 37 44 49 53 58 63 67 72 79
## 1683  0  1  0  1  3  2  1  0  0  2  0  1  0  0  0  1
## 1688 15  1  0  1  0  0  0  0  0  0  0  0  0  0  0  1
## 1692  0 24 24  0 24 20  6  0  1  8 16  0  0  6  0  0
## 1699  0  0 24  0  1  0  0  0  1  0  0  0  1  0  0  0
```

In this dataset the counts are all zero, so no genome (including Turicimonas) introduces missing allele-frequency entries.

7. Treatment groups

```
unique(long_df$treatment_group)
```

```
## [1] "Control"       "Ciprofloxacin" "Tetracyclin"   "Vancomycin"
```

```
table(long_df$treatment_group)
```

```
##
## Ciprofloxacin       Control   Tetracyclin    Vancomycin
##           768           768           768           768
```

```r
table(long_df$mouse_id, long_df$treatment_group)
```

```
##
##        Ciprofloxacin Control Tetracyclin Vancomycin
##   1683             0     768           0          0
##   1688           768       0           0          0
##   1692             0       0         768          0
##   1699             0       0           0        768
```

```r
long_df[long_df$day == 30, c('mouse_id','treatment_group')]
```

```
##      mouse_id treatment_group
## 193      1683         Control
## 194      1683         Control
## 195      1683         Control
## 196      1683         Control
## 197      1683         Control
## 198      1683         Control
## 199      1683         Control
## 200      1683         Control
## 201      1683         Control
## 202      1683         Control
## 203      1683         Control
## 204      1683         Control
## 205      1683         Control
## 206      1683         Control
## 207      1683         Control
## 208      1683         Control
## 209      1683         Control
## 210      1683         Control
## 211      1683         Control
## 212      1683         Control
## 213      1683         Control
## 214      1683         Control
## 215      1683         Control
## 216      1683         Control
## 217      1683         Control
## 218      1683         Control
## 219      1683         Control
## 220      1683         Control
## 221      1683         Control
## 222      1683         Control
## 223      1683         Control
## 224      1683         Control
## 225      1683         Control
## 226      1683         Control
## 227      1683         Control
## 228      1683         Control
## 229      1683         Control
## 230      1683         Control
## 231      1683         Control
## 232      1683         Control
## 233      1683         Control
## 234      1683         Control
```

```
## 235      1683         Control
## 236      1683         Control
## 237      1683         Control
## 238      1683         Control
## 239      1683         Control
## 240      1683         Control
## 961      1688   Ciprofloxacin
## 962      1688   Ciprofloxacin
## 963      1688   Ciprofloxacin
## 964      1688   Ciprofloxacin
## 965      1688   Ciprofloxacin
## 966      1688   Ciprofloxacin
## 967      1688   Ciprofloxacin
## 968      1688   Ciprofloxacin
## 969      1688   Ciprofloxacin
## 970      1688   Ciprofloxacin
## 971      1688   Ciprofloxacin
## 972      1688   Ciprofloxacin
## 973      1688   Ciprofloxacin
## 974      1688   Ciprofloxacin
## 975      1688   Ciprofloxacin
## 976      1688   Ciprofloxacin
## 977      1688   Ciprofloxacin
## 978      1688   Ciprofloxacin
## 979      1688   Ciprofloxacin
## 980      1688   Ciprofloxacin
## 981      1688   Ciprofloxacin
## 982      1688   Ciprofloxacin
## 983      1688   Ciprofloxacin
## 984      1688   Ciprofloxacin
## 985      1688   Ciprofloxacin
## 986      1688   Ciprofloxacin
## 987      1688   Ciprofloxacin
## 988      1688   Ciprofloxacin
## 989      1688   Ciprofloxacin
## 990      1688   Ciprofloxacin
## 991      1688   Ciprofloxacin
## 992      1688   Ciprofloxacin
## 993      1688   Ciprofloxacin
## 994      1688   Ciprofloxacin
## 995      1688   Ciprofloxacin
## 996      1688   Ciprofloxacin
## 997      1688   Ciprofloxacin
## 998      1688   Ciprofloxacin
## 999      1688   Ciprofloxacin
## 1000     1688   Ciprofloxacin
## 1001     1688   Ciprofloxacin
## 1002     1688   Ciprofloxacin
## 1003     1688   Ciprofloxacin
## 1004     1688   Ciprofloxacin
## 1005     1688   Ciprofloxacin
## 1006     1688   Ciprofloxacin
## 1007     1688   Ciprofloxacin
## 1008     1688   Ciprofloxacin
```

```
## 1729     1692     Tetracyclin
## 1730     1692     Tetracyclin
## 1731     1692     Tetracyclin
## 1732     1692     Tetracyclin
## 1733     1692     Tetracyclin
## 1734     1692     Tetracyclin
## 1735     1692     Tetracyclin
## 1736     1692     Tetracyclin
## 1737     1692     Tetracyclin
## 1738     1692     Tetracyclin
## 1739     1692     Tetracyclin
## 1740     1692     Tetracyclin
## 1741     1692     Tetracyclin
## 1742     1692     Tetracyclin
## 1743     1692     Tetracyclin
## 1744     1692     Tetracyclin
## 1745     1692     Tetracyclin
## 1746     1692     Tetracyclin
## 1747     1692     Tetracyclin
## 1748     1692     Tetracyclin
## 1749     1692     Tetracyclin
## 1750     1692     Tetracyclin
## 1751     1692     Tetracyclin
## 1752     1692     Tetracyclin
## 1753     1692     Tetracyclin
## 1754     1692     Tetracyclin
## 1755     1692     Tetracyclin
## 1756     1692     Tetracyclin
## 1757     1692     Tetracyclin
## 1758     1692     Tetracyclin
## 1759     1692     Tetracyclin
## 1760     1692     Tetracyclin
## 1761     1692     Tetracyclin
## 1762     1692     Tetracyclin
## 1763     1692     Tetracyclin
## 1764     1692     Tetracyclin
## 1765     1692     Tetracyclin
## 1766     1692     Tetracyclin
## 1767     1692     Tetracyclin
## 1768     1692     Tetracyclin
## 1769     1692     Tetracyclin
## 1770     1692     Tetracyclin
## 1771     1692     Tetracyclin
## 1772     1692     Tetracyclin
## 1773     1692     Tetracyclin
## 1774     1692     Tetracyclin
## 1775     1692     Tetracyclin
## 1776     1692     Tetracyclin
## 2497     1699      Vancomycin
## 2498     1699      Vancomycin
## 2499     1699      Vancomycin
## 2500     1699      Vancomycin
## 2501     1699      Vancomycin
## 2502     1699      Vancomycin
```

```
## 2503     1699      Vancomycin
## 2504     1699      Vancomycin
## 2505     1699      Vancomycin
## 2506     1699      Vancomycin
## 2507     1699      Vancomycin
## 2508     1699      Vancomycin
## 2509     1699      Vancomycin
## 2510     1699      Vancomycin
## 2511     1699      Vancomycin
## 2512     1699      Vancomycin
## 2513     1699      Vancomycin
## 2514     1699      Vancomycin
## 2515     1699      Vancomycin
## 2516     1699      Vancomycin
## 2517     1699      Vancomycin
## 2518     1699      Vancomycin
## 2519     1699      Vancomycin
## 2520     1699      Vancomycin
## 2521     1699      Vancomycin
## 2522     1699      Vancomycin
## 2523     1699      Vancomycin
## 2524     1699      Vancomycin
## 2525     1699      Vancomycin
## 2526     1699      Vancomycin
## 2527     1699      Vancomycin
## 2528     1699      Vancomycin
## 2529     1699      Vancomycin
## 2530     1699      Vancomycin
## 2531     1699      Vancomycin
## 2532     1699      Vancomycin
## 2533     1699      Vancomycin
## 2534     1699      Vancomycin
## 2535     1699      Vancomycin
## 2536     1699      Vancomycin
## 2537     1699      Vancomycin
## 2538     1699      Vancomycin
## 2539     1699      Vancomycin
## 2540     1699      Vancomycin
## 2541     1699      Vancomycin
## 2542     1699      Vancomycin
## 2543     1699      Vancomycin
## 2544     1699      Vancomycin
```

8. Stretch idea

```
day30 <- long_df[long_df$day == 30, ]
medians_day30 <- tapply(day30$value, day30$Genome, median, na.rm = TRUE)
medians_day30
```

```
## Akkermansia_muciniphila_YL44   Bacteroides_caecimuris_I48
##                     0.124955                     0.030887
##       Turicimonas_muris_YL45
##                     0.250000
```

```
medians_day30[which.max(medians_day30)]
```

```
## Turicimonas_muris_YL45
##                   0.25
```

Feel free to compare your answers with these outputs, then proceed to `scripts/individual_notebooks/02_simple_heatmap`

Solution – Simple Heatmap

Below is one possible solution for the worksheet that builds the heatmap from `dataset2_subset.csv` / `dataset2_subset_long.csv` (three genomes). Feel free to compare this with your own approach.

1. Load packages and define paths

```
library(ComplexHeatmap)
```

```
## Loading required package: grid
```

```
## ========================================
## ComplexHeatmap version 2.24.1
## Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
## Github page: https://github.com/jokergoo/ComplexHeatmap
## Documentation: http://jokergoo.github.io/ComplexHeatmap-reference
##
## If you use it in published research, please cite either one:
## - Gu, Z. Complex Heatmap Visualization. iMeta 2022.
## - Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
##     genomic data. Bioinformatics 2016.
##
##
## The new InteractiveComplexHeatmap package can directly export static
## complex heatmaps into an interactive Shiny app with zero effort. Have a try!
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(ComplexHeatmap))
## ========================================
```

```
library(circlize)
```

```
## ========================================
## circlize version 0.4.16
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: https://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
##   in R. Bioinformatics 2014.
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(circlize))
## ========================================
```

```
subset_path <- file.path('..', 'data', 'dataset2_subset.csv')
long_path <- file.path('..', 'data', 'dataset2_subset_long.csv')
```

```
pdf_path <- file.path('..', 'pdf', 'dataset2_heatmap.pdf')
```

2. Load/inspect the data

```
wide_df <- read.csv(subset_path, check.names = FALSE, stringsAsFactors = FALSE)
long_df <- read.csv(long_path, check.names = FALSE, stringsAsFactors = FALSE)

cat('Wide table dimensions:', nrow(wide_df), 'rows x', ncol(wide_df), 'columns\n')
```

```
## Wide table dimensions: 48 rows x 67 columns
```

```
cat('Long table dimensions:', nrow(long_df), 'rows x', ncol(long_df), 'columns\n')
```

```
## Long table dimensions: 3072 rows x 7 columns
```

```
head(wide_df[, c('Genome', 'snp_id', 'Position')])
```

```
##                          Genome      snp_id Position
## 1 Akkermansia_muciniphila_YL44  239840-C-G   239840
## 2 Akkermansia_muciniphila_YL44  241793-A-G   241793
## 3 Akkermansia_muciniphila_YL44  355328-A-T   355328
## 4 Akkermansia_muciniphila_YL44  356291-C-A   356291
## 5 Akkermansia_muciniphila_YL44 2351445-C-T  2351445
## 6   Bacteroides_caecimuris_I48 1601848-T-C  1601848
```

```
head(long_df)
```

```
##                          Genome      snp_id Position    value mouse_id day
## 1 Akkermansia_muciniphila_YL44  239840-C-G   239840 0.000000     1683   0
## 2 Akkermansia_muciniphila_YL44  241793-A-G   241793 0.049587     1683   0
## 3 Akkermansia_muciniphila_YL44  355328-A-T   355328 0.138182     1683   0
## 4 Akkermansia_muciniphila_YL44  356291-C-A   356291 0.000000     1683   0
## 5 Akkermansia_muciniphila_YL44 2351445-C-T  2351445 0.000000     1683   0
## 6   Bacteroides_caecimuris_I48 1601848-T-C  1601848 0.041609     1683   0
##   treatment_group
## 1         Control
## 2         Control
## 3         Control
## 4         Control
## 5         Control
## 6         Control
```

3. Choose a treatment group subset

```
target_group <- 'Control'
sample_meta <- unique(long_df[, c('mouse_id', 'day', 'treatment_group')])
sample_meta$sample_id <- paste(sample_meta$mouse_id, sample_meta$day, sep = '-')
keep_samples <- sample_meta$sample_id[sample_meta$treatment_group == target_group]
wide_df <- wide_df[, c('Genome', 'snp_id', 'Position', keep_samples)]
```

4. Quick summaries

```
print(table(wide_df$Genome))
```

```
##
```

```
## Akkermansia_muciniphila_YL44   Bacteroides_caecimuris_I48
##                            5                           19
##       Turicimonas_muris_YL45
##                           24
```

```
mouse_day_table <- with(long_df, table(mouse_id, day))
print(mouse_day_table)
```

```
##          day
## mouse_id  0  4  9 14 18 23 30 37 44 49 53 58 63 67 72 79
##     1683 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48
##     1688 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48
##     1692 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48
##     1699 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48
```

```
value_summary <- tapply(long_df$value, long_df$Genome, function(x) {
  c(min = min(x, na.rm = TRUE),
    median = median(x, na.rm = TRUE),
    max = max(x, na.rm = TRUE))
})
value_summary <- do.call(rbind, value_summary)
print(value_summary)
```

```
##                              min    median      max
## Akkermansia_muciniphila_YL44   0 0.0523090 0.871245
## Bacteroides_caecimuris_I48     0 0.0338305 0.991218
## Turicimonas_muris_YL45         0 0.2649125 1.000000
```

5. Build the heatmap matrix

```
sample_cols <- setdiff(names(wide_df), c('Genome', 'snp_id', 'Position'))
heatmap_matrix <- as.matrix(wide_df[, sample_cols])
mode(heatmap_matrix) <- 'numeric'
rownames(heatmap_matrix) <- paste(wide_df$Genome, wide_df$snp_id, sep = ' | ')

sample_meta <- data.frame(sample_id = sample_cols, stringsAsFactors = FALSE)
split_ids <- strsplit(sample_meta$sample_id, '-', fixed = TRUE)
sample_meta$mouse_id <- vapply(split_ids, function(x) x[[1]], character(1))
sample_meta$day <- as.integer(vapply(split_ids, function(x) if (length(x) >= 2) x[[2]] else NA_characte

order_idx <- order(sample_meta$mouse_id, sample_meta$day, sample_meta$sample_id)
sample_meta <- sample_meta[order_idx, ]
heatmap_matrix <- heatmap_matrix[, sample_meta$sample_id, drop = FALSE]
```

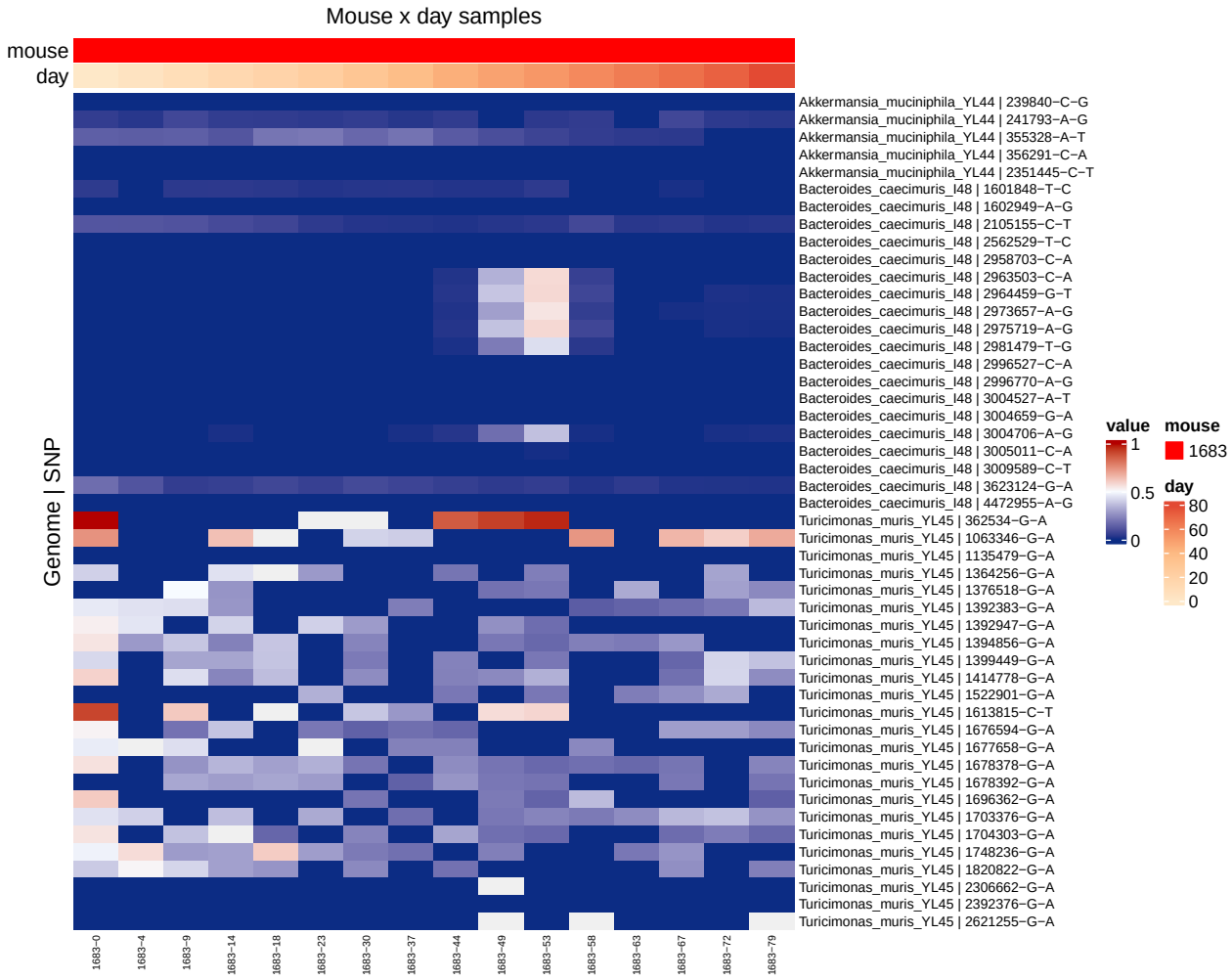
6. Colors and annotations

```
min_val <- min(heatmap_matrix, na.rm = TRUE)
max_val <- max(heatmap_matrix, na.rm = TRUE)
if (!is.finite(min_val)) min_val <- 0
if (!is.finite(max_val)) max_val <- 1
if (abs(max_val - min_val) < .Machine$double.eps) {
  max_val <- min_val + 1
}
mid_val <- (min_val + max_val) / 2
```

```r
color_fun <- circlize::colorRamp2(c(min_val, mid_val, max_val),
                                  c('#0c2c84', '#f7fbff', '#b30000'))

mouse_levels <- unique(sample_meta$mouse_id)
mouse_colors <- setNames(grDevices::rainbow(length(mouse_levels)), mouse_levels)

min_day <- min(sample_meta$day, na.rm = TRUE)
max_day <- max(sample_meta$day, na.rm = TRUE)
if (min_day == max_day) {
  day_colors <- circlize::colorRamp2(c(min_day, min_day + 1), c('#fee8c8', '#e34a33'))
} else {
  day_colors <- circlize::colorRamp2(seq(min_day, max_day, length.out = 3),
                                     c('#fee8c8', '#fdbb84', '#e34a33'))
}

col_annotation <- HeatmapAnnotation(
  mouse = factor(sample_meta$mouse_id, levels = mouse_levels),
  day = sample_meta$day,
  col = list(mouse = mouse_colors, day = day_colors),
  annotation_name_side = 'left'
)
```

7. Draw and export the heatmap

```r
row_name_size <- max(5, min(8, 40 / log10(max(10, nrow(heatmap_matrix)))))
col_name_size <- max(6, min(10, 80 / ncol(heatmap_matrix)))

ht <- Heatmap(
  heatmap_matrix,
  name = 'value',
  col = color_fun,
  na_col = '#f0f0f0',
  top_annotation = col_annotation,
  column_split = factor(sample_meta$mouse_id, levels = mouse_levels),
  cluster_rows = FALSE,
  cluster_columns = FALSE,
  column_title = 'Mouse x day samples',
  row_title = 'Genome | SNP',
  show_row_names = TRUE,
  show_column_names = TRUE,
  row_names_gp = grid::gpar(fontsize = row_name_size),
  column_names_gp = grid::gpar(fontsize = col_name_size)
)

draw(ht, heatmap_legend_side = 'right', annotation_legend_side = 'right')
```




```
pdf_height <- max(6, min(18, 0.2 * nrow(heatmap_matrix) + 4))
pdf_width <- max(8, min(16, 0.2 * ncol(heatmap_matrix) + 6))

dir.create(dirname(pdf_path), recursive = TRUE, showWarnings = FALSE)
pdf(pdf_path, width = pdf_width, height = pdf_height)
draw(ht, heatmap_legend_side = 'right', annotation_legend_side = 'right')
dev.off()
```

```
## pdf
##   2
```

```
cat('Saved heatmap to', pdf_path, '\n')
```

```
## Saved heatmap to ../pdf/dataset2_heatmap.pdf
```

For extra practice, try adding `row_split` by Genome or experiment with a different color palette.

Solution – Full Heatmap

This solution notebook offers one way to complete the full heatmap exercise. Feel free to tweak palettes, ordering, or filtering thresholds to suit your teaching needs.

1. Packages, paths, helpers

```
suppressPackageStartupMessages({
  library(ComplexHeatmap)
  library(circlize)
  library(viridisLite)
})
subset_path <- file.path('..','data','dataset3_subset.csv')
long_path   <- file.path('..','data','dataset3_subset_long.csv')
pdf_path    <- file.path('..','pdf','04_full_heatmap_exercise.pdf')
na_color <- '#dcdcdc'
```

2. Load data and NA report

```
wide_df <- read.csv(subset_path, check.names = FALSE, stringsAsFactors = FALSE)
long_df <- read.csv(long_path, check.names = FALSE, stringsAsFactors = FALSE)
cat('Wide rows x cols:', nrow(wide_df), ncol(wide_df), '\n')
```

```
## Wide rows x cols: 71 67
```

```
cat('Long rows x cols:', nrow(long_df), ncol(long_df), '\n')
```

```
## Long rows x cols: 4544 7
```

```
na_total <- sum(is.na(long_df$value))
cat('NA count (value):', na_total, '\n')
```

```
## NA count (value): 443
```

```
if (na_total > 0) {
  na_table <- with(long_df, tapply(value, list(mouse_id, day), function(x) sum(is.na(x))))
  print(na_table)
}
```

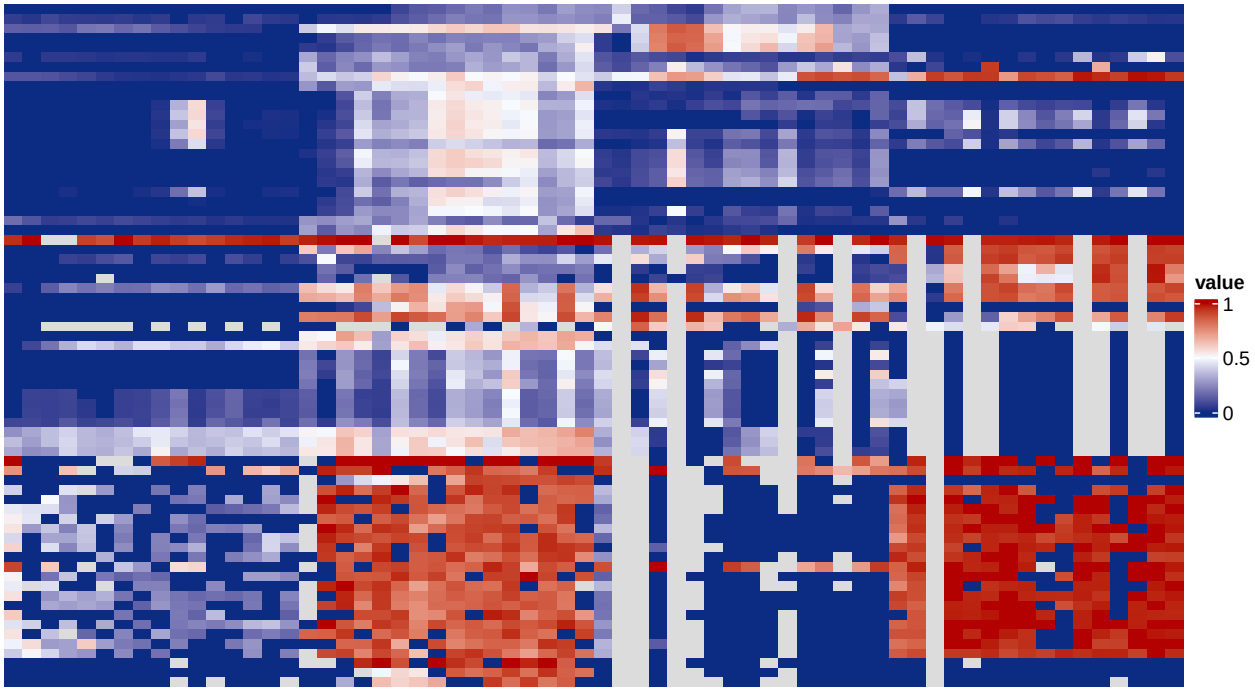
```
##       0  4  9 14 18 23 30 37 44 49 53 58 63 67 72 79
## 1683  0  1  2  3  4  4  2  0  1  2  1  1  1  0  1  1
## 1688 15  1  1  2  3  0  1  1  0  0  0  1  0  0  1  1
## 1692  0 47 24  0 46 20  6  0  1  9 38  0  0 28  0  0
## 1699  0 23 37  0 24 13  0  0  1  1 23 13  1 23 13  1
```

3. Matrix + metadata

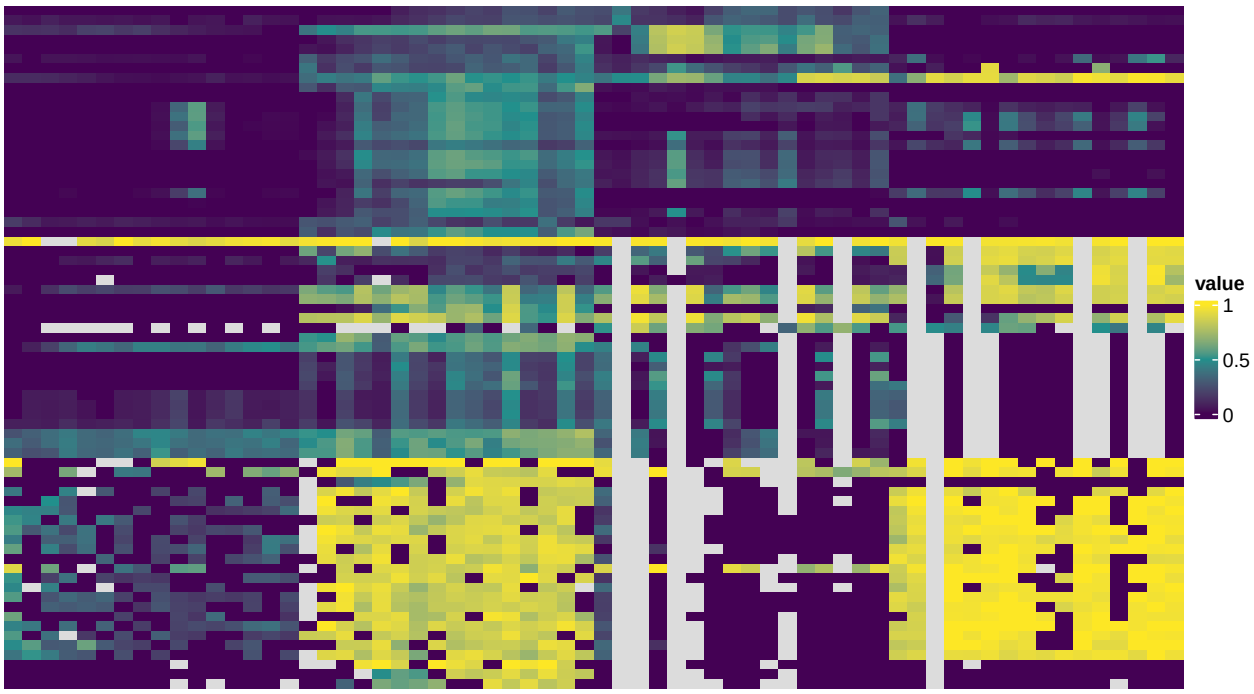
```
sample_cols <- setdiff(names(wide_df), c('Genome','snp_id','Position'))
mat <- as.matrix(wide_df[, sample_cols])
mode(mat) <- 'numeric'
rownames(mat) <- paste(wide_df$Genome, wide_df$snp_id, sep = ' | ')

sample_meta <- unique(long_df[, c('mouse_id','day','treatment_group')])
sample_meta$sample_id <- paste(sample_meta$mouse_id, sample_meta$day, sep='-')
sample_meta <- sample_meta[match(colnames(mat), sample_meta$sample_id), ]
stopifnot(identical(colnames(mat), sample_meta$sample_id))
```

4. Baseline heatmap + palettes

```
mins <- min(mat, na.rm = TRUE)
maxs <- max(mat, na.rm = TRUE)
mids <- (mins + maxs) / 2
palette_blue_red <- circlize::colorRamp2(c(mins, mids, maxs), c('#0c2c84','#f7fbff','#b30000'))
palette_viridis <- circlize::colorRamp2(c(mins, mids, maxs), viridisLite::viridis(3))

ht_blue <- Heatmap(mat, name = 'value', col = palette_blue_red, na_col = na_color,
                   cluster_rows = FALSE, cluster_columns = FALSE,
                   show_row_names = FALSE, show_column_names = FALSE)
ht_viridis <- Heatmap(mat, name = 'value', col = palette_viridis, na_col = na_color,
                      cluster_rows = FALSE, cluster_columns = FALSE,
                      show_row_names = FALSE, show_column_names = FALSE)

ht_blue
```

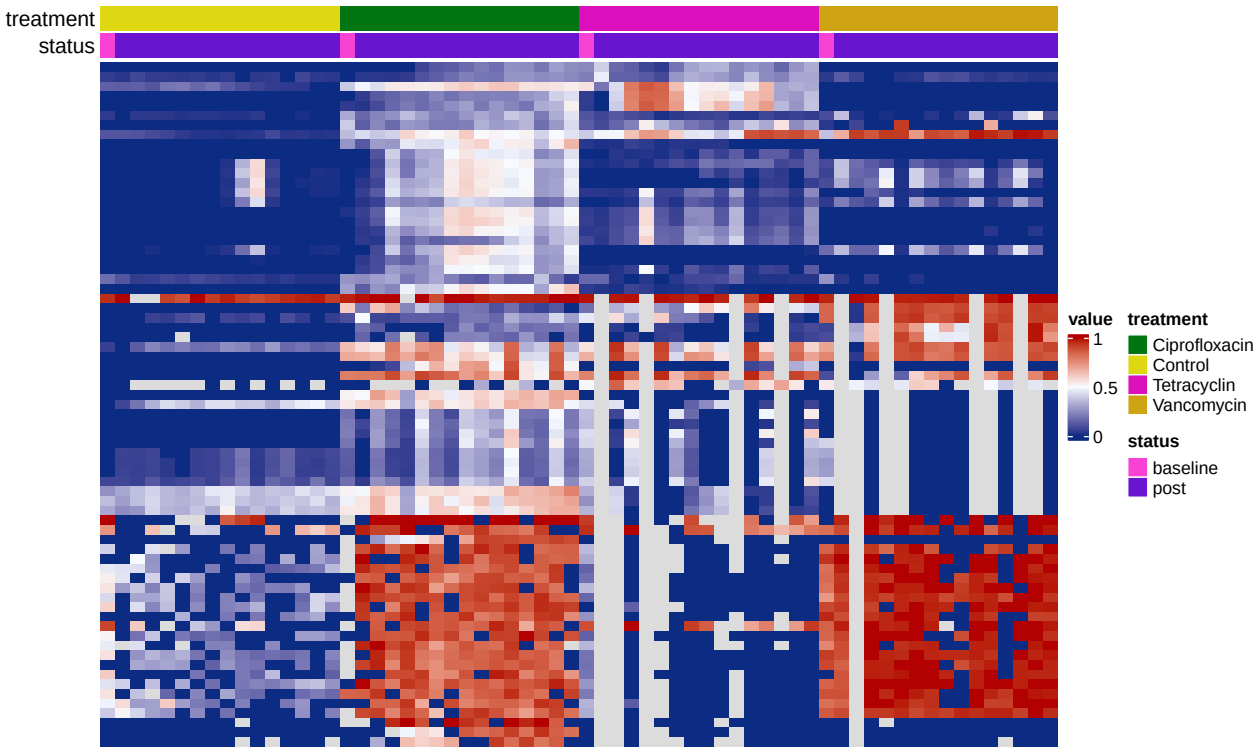



```
ht_viridis
```



5. Annotations + ordering

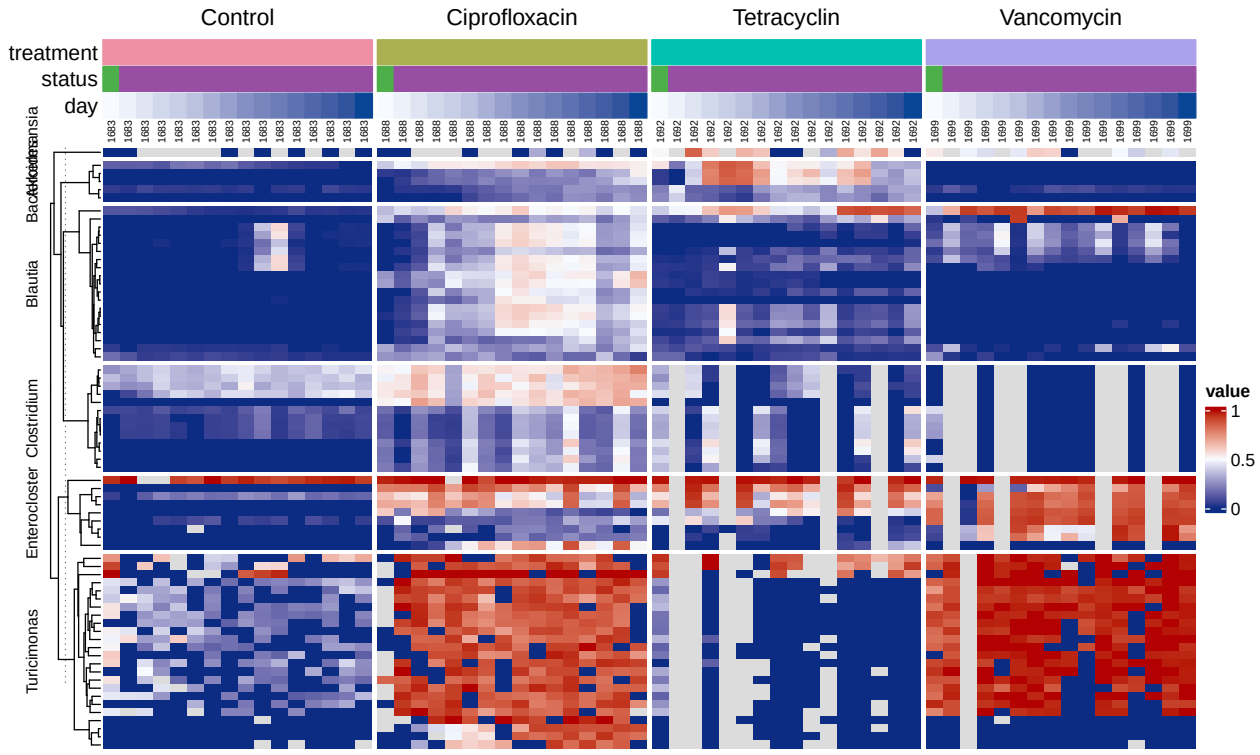
```
sample_meta$post_ab <- ifelse(sample_meta$day == 0, 'baseline', 'post')
col_ann <- HeatmapAnnotation(
  treatment = sample_meta$treatment_group,
  status = sample_meta$post_ab,
  annotation_name_side = 'left'
)
order_idx <- order(sample_meta$mouse_id, sample_meta$day)
mat_ordered <- mat[, order_idx]
col_ann_ordered <- col_ann[order_idx]

Heatmap(
  mat_ordered,
  name = 'value',
  col = palette_blue_red,
  top_annotation = col_ann_ordered,
  cluster_rows = FALSE,
  cluster_columns = FALSE,
  show_row_names = FALSE,
  show_column_names = FALSE,
  na_col = na_color
)
```



6. Annotation enhancements

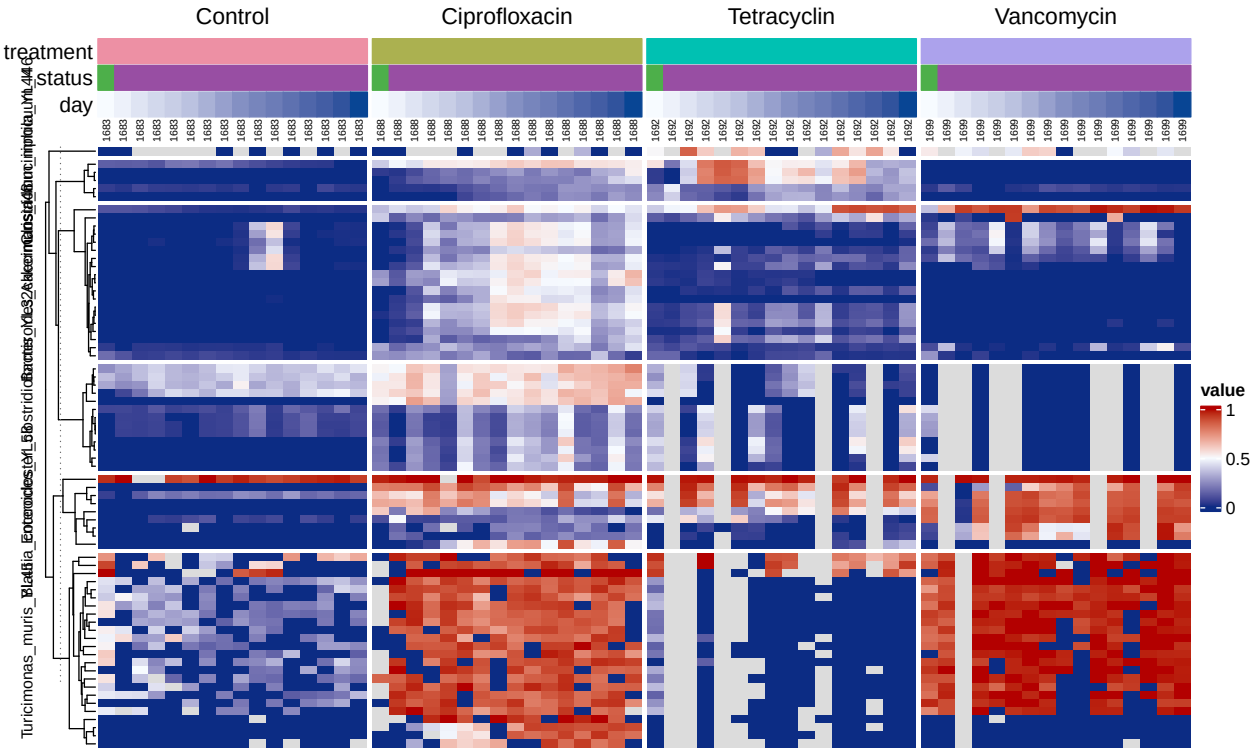
```
treatment_levels <- unique(sample_meta$treatment_group)
treatment_cols <- grDevices::hcl.colors(length(treatment_levels), palette = "Set2")
names(treatment_cols) <- treatment_levels
status_cols <- c(baseline = '#4daf4a', post = '#984ea3')
day_col_fun <- circlize::colorRamp2(range(sample_meta$day), c('#f7fbff','#084594'))
col_ann_rich <- HeatmapAnnotation(
  treatment = anno_simple(sample_meta$treatment_group, col = treatment_cols),
  status = anno_simple(sample_meta$post_ab, col = status_cols),
  day = anno_simple(sample_meta$day, col = day_col_fun),
  mouse = anno_text(sample_meta$mouse_id, rot = 90, gp = grid::gpar(fontsize = 6)),
  annotation_name_side = 'left'
)
col_ann_rich_ordered <- col_ann_rich[order_idx]
column_split <- factor(sample_meta$treatment_group[order_idx], levels = treatment_levels)
row_split <- factor(wide_df$Genome, levels = unique(wide_df$Genome))
row_titles <- sub('_.*', '', levels(row_split))

Heatmap(
  mat_ordered,
  name = 'value',
  col = palette_blue_red,
  top_annotation = col_ann_rich_ordered,
  cluster_rows = TRUE,
  cluster_columns = FALSE,
  column_split = column_split,
  show_row_names = FALSE,
  show_column_names = FALSE,
```

```
  na_col = na_color,
  row_split = row_split,
  row_title = row_titles,
  row_title_gp = grid::gpar(fontsize = 9)
)
```



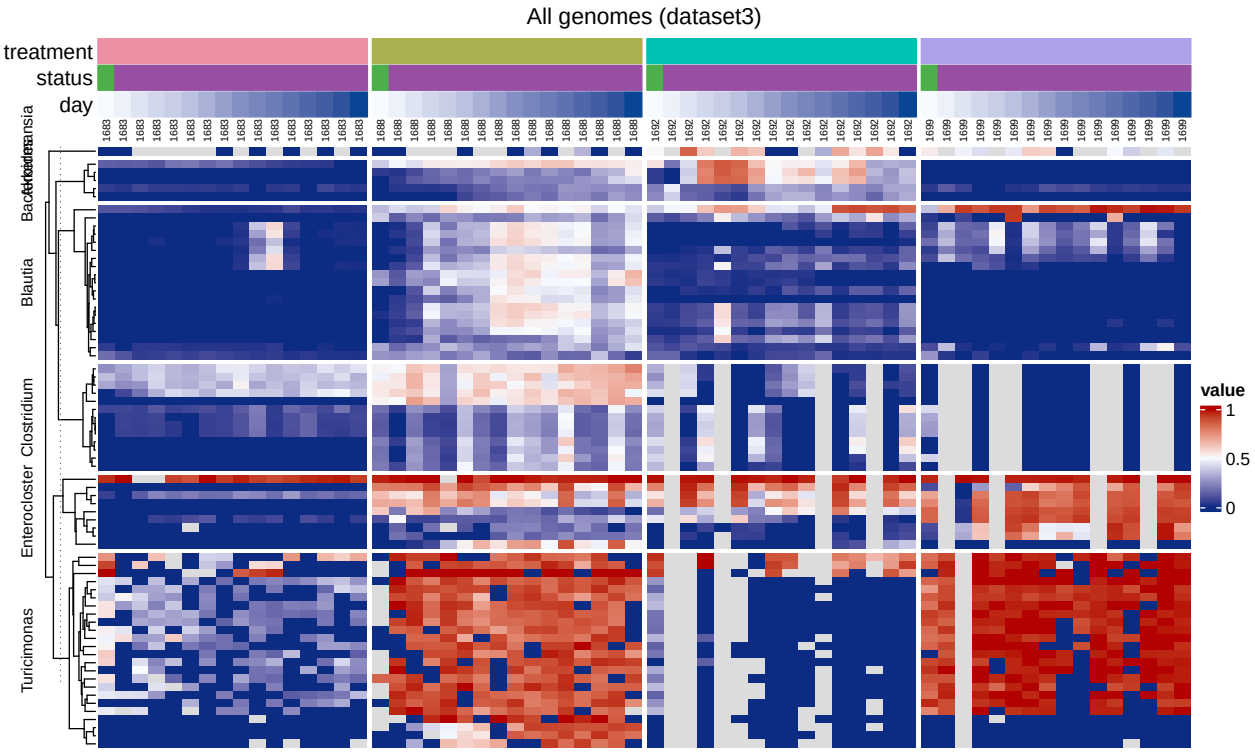
7. Row variance filter

```
row_var <- apply(mat, 1, var, na.rm = TRUE)
keep_idx <- order(row_var, decreasing = TRUE)[seq_len(min(100, nrow(mat)))]
mat_topvar <- mat[keep_idx, order_idx]
row_split_top <- droplevels(row_split[keep_idx])
Heatmap(
  mat_topvar,
  name = 'value',
  col = palette_blue_red,
  top_annotation = col_ann_rich_ordered,
  cluster_rows = TRUE,
  cluster_columns = FALSE,
  column_split = column_split,
  show_row_names = FALSE,
  show_column_names = FALSE,
  na_col = na_color,
  row_split = row_split_top,
  row_title_gp = grid::gpar(fontsize = 9)
)
```



8. Final heatmap + PDF export

```
ht_final <- Heatmap(
  mat_ordered,
  name = 'value',
  col = palette_blue_red,
  top_annotation = col_ann_rich_ordered,
  cluster_rows = TRUE,
  cluster_columns = FALSE,
  column_split = column_split,
  show_row_names = FALSE,
  show_column_names = FALSE,
  na_col = na_color,
  row_split = row_split,
  row_title = row_titles,
  row_title_gp = grid::gpar(fontsize = 9),
  column_title = 'All genomes (dataset3)'
)
draw(ht_final)
```




```
pdf(pdf_path, width = 11, height = 7)
draw(ht_final)
dev.off()
cat('Saved heatmap to', pdf_path, '\n')
```

9. Notes

- Palette choice: the blue-white-red ramp emphasizes deviations from mid values.
- Column ordering by mouse/day reveals antibiotic pulses more clearly than the CSV default.
- Variance filtering is helpful when presenting in class; it trims the figure to the most dynamic SNPs and speeds up PDF export.
- Extra annotations (day gradient + mouse labels + row splits) plus column splits by treatment mirror the annotation tutorial and keep the story tied to the experimental design.

Solution – Bonus Heatmap Customization

This key demonstrates one way to complete the bonus exercise combining decorations, legends, and a multi-heatmap layout for the dataset3 tables.

1. Setup and metadata

```
suppressPackageStartupMessages({
  library(ComplexHeatmap)
  library(circlize)
  library(viridisLite)
  library(grid)
})

subset_path <- file.path('..', 'data', 'dataset3_subset.csv')
```

```r
long_path   <- file.path('..', 'data', 'dataset3_subset_long.csv')
wide_df <- read.csv(subset_path, check.names = FALSE, stringsAsFactors = FALSE)
long_df <- read.csv(long_path, check.names = FALSE, stringsAsFactors = FALSE)

long_df$sample_id <- paste(long_df$mouse_id, long_df$day, sep = '-')
sample_meta <- unique(long_df[, c('sample_id', 'mouse_id', 'day', 'treatment_group')])
sample_meta$day <- as.integer(sample_meta$day)

# Compare two treatment groups for this exercise
target_groups <- c('Control', 'Ciprofloxacin')
sample_meta <- sample_meta[sample_meta$treatment_group %in% target_groups, ]
order_idx <- order(sample_meta$treatment_group, sample_meta$mouse_id, sample_meta$day)
sample_meta <- sample_meta[order_idx, ]
row.names(sample_meta) <- NULL
```

2. Matrix, row variance, and palette

```r
sample_cols <- sample_meta$sample_id
wide_subset <- wide_df[, c('Genome', 'snp_id', 'Position', sample_cols)]
heatmap_matrix <- as.matrix(wide_subset[, sample_cols])
mode(heatmap_matrix) <- 'numeric'
rownames(heatmap_matrix) <- paste(wide_subset$Genome, wide_subset$snp_id, sep = ' | ')
row_genome <- wide_subset$Genome
row_var <- apply(heatmap_matrix, 1, var, na.rm = TRUE)

value_range <- range(heatmap_matrix, na.rm = TRUE)
color_fun <- circlize::colorRamp2(
  seq(value_range[1], value_range[2], length.out = 5),
  viridisLite::viridis(5)
)
```

3. Annotation summaries

```r
print(table(sample_meta$treatment_group))
```

```
##
## Ciprofloxacin       Control
##            16            16
```

```r
print(table(sample_meta$treatment_group, sample_meta$day))
```

```
##
##                 0 4 9 14 18 23 30 37 44 49 53 58 63 67 72 79
##   Ciprofloxacin 1 1 1  1  1  1  1  1  1  1  1  1  1  1  1  1
##   Control       1 1 1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

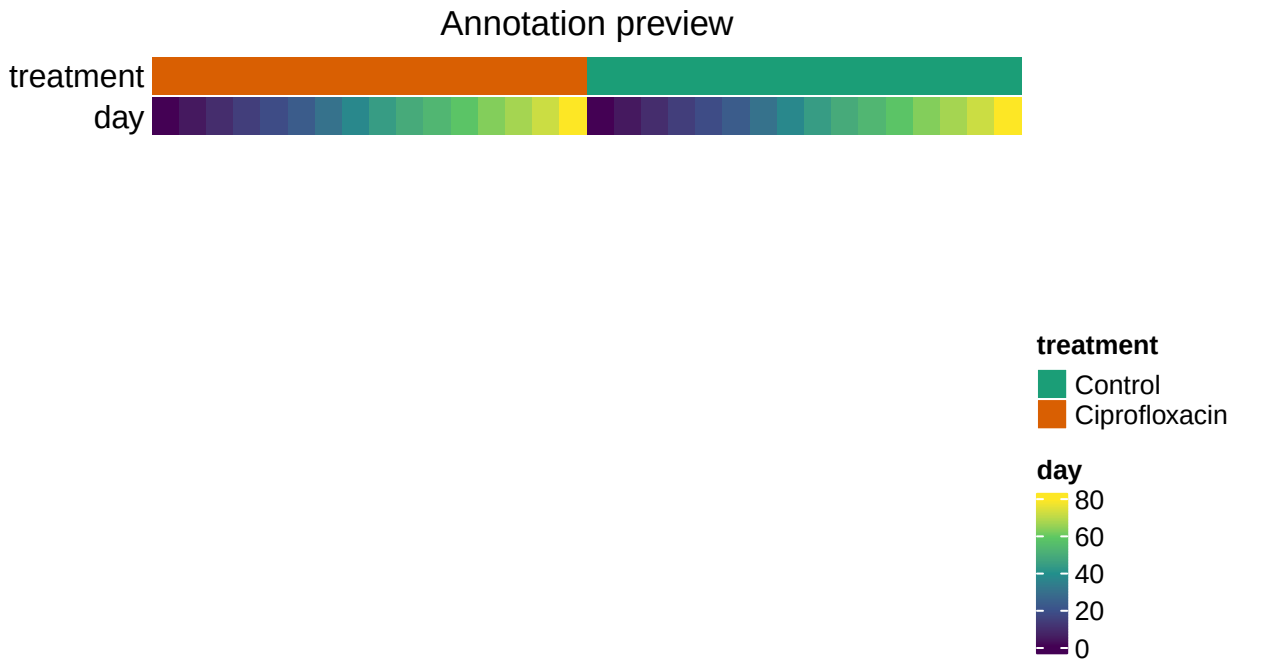
```r
head(sample_meta)
```

```
##   sample_id mouse_id day treatment_group
## 1    1688-0     1688   0   Ciprofloxacin
## 2    1688-4     1688   4   Ciprofloxacin
## 3    1688-9     1688   9   Ciprofloxacin
## 4   1688-14     1688  14   Ciprofloxacin
## 5   1688-18     1688  18   Ciprofloxacin
```

```
## 6   1688-23     1688  23   Ciprofloxacin
```

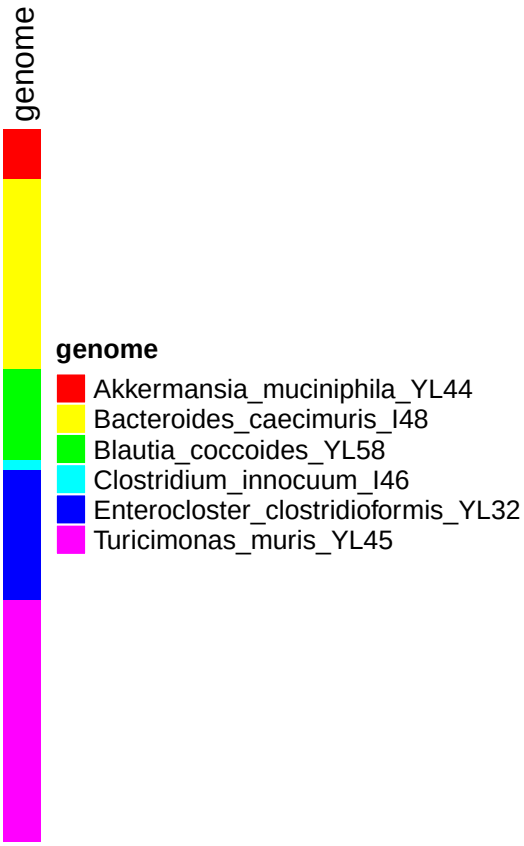
4. Column and row annotations

```
treatment_colors <- setNames(c('#1b9e77', '#d95f02'), target_groups)
day_seq <- seq(min(sample_meta$day), max(sample_meta$day), length.out = 5)
day_colors <- circlize::colorRamp2(day_seq, viridisLite::viridis(5))

col_ann <- HeatmapAnnotation(
  treatment = factor(sample_meta$treatment_group, levels = target_groups),
  day = sample_meta$day,
  col = list(treatment = treatment_colors, day = day_colors),
  annotation_name_side = 'left'
)

genome_levels <- unique(row_genome)
genome_colors <- setNames(grDevices::rainbow(length(genome_levels)), genome_levels)
row_ann <- rowAnnotation(
  genome = factor(row_genome, levels = genome_levels),
  col = list(genome = genome_colors),
  annotation_name_side = 'top'
)

preview_mat <- matrix(0, nrow = 1, ncol = ncol(heatmap_matrix))
preview_ht <- Heatmap(
  preview_mat,
  col = c('0' = '#ffffff'),
  top_annotation = col_ann,
  cluster_rows = FALSE,
  cluster_columns = FALSE,
  show_heatmap_legend = FALSE,
  show_row_names = FALSE,
  show_column_names = FALSE,
  column_title = 'Annotation preview'
)

draw(preview_ht, annotation_legend_side = 'right')
```




```
row_preview <- Heatmap(
  matrix(0, nrow = nrow(heatmap_matrix), ncol = 1),
  col = c('0' = '#ffffff'),
  right_annotation = row_ann,
  cluster_rows = FALSE,
  cluster_columns = FALSE,
  show_row_names = FALSE,
  show_column_names = FALSE,
  show_heatmap_legend = FALSE,
  column_title = 'Row annotation preview'
)

draw(row_preview, heatmap_legend_side = 'right', annotation_legend_side = 'right')
```

Row annotation preview



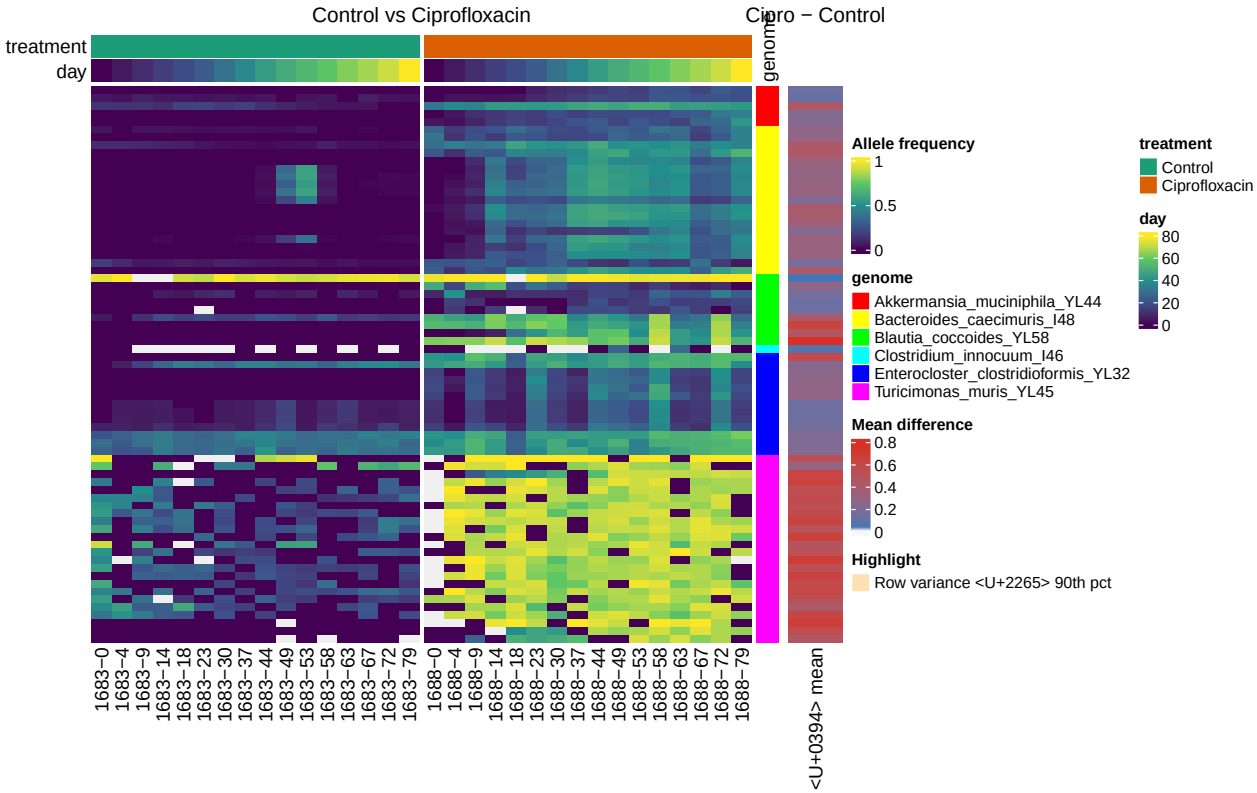
5. Heatmap for high-variance rows

```
high_var_cut <- quantile(row_var, 0.9, na.rm = TRUE)
highlight_rows <- which(row_var >= high_var_cut)

main_ht <- Heatmap(
  heatmap_matrix,
  name = 'bonus_heatmap',
  col = color_fun,
  top_annotation = col_ann,
  right_annotation = row_ann,
  column_split = factor(sample_meta$treatment_group, levels = target_groups),
  cluster_rows = FALSE,
  cluster_columns = FALSE,
  show_row_names = FALSE,
  column_title = 'Control vs Ciprofloxacin',
  heatmap_legend_param = list(title = 'Allele frequency'),
  na_col = '#f0f0f0'
)
```

6. Companion heatmap, custom legend, and draw call

```
control_cols <- sample_meta$sample_id[sample_meta$treatment_group == target_groups[1]]
treated_cols <- sample_meta$sample_id[sample_meta$treatment_group == target_groups[2]]
control_mean <- rowMeans(heatmap_matrix[, control_cols, drop = FALSE], na.rm = TRUE)
treated_mean <- rowMeans(heatmap_matrix[, treated_cols, drop = FALSE], na.rm = TRUE)
```

```
mean_delta <- treated_mean - control_mean

delta_ht <- Heatmap(
  mean_delta,
  name = 'Δ mean',
  width = unit(1.2, 'cm'),
  col = circlize::colorRamp2(
    c(min(mean_delta, na.rm = TRUE), 0, max(mean_delta, na.rm = TRUE)),
    c('#4575b4', '#f7f7f7', '#d73027')
  ),
  show_row_names = FALSE,
  cluster_rows = FALSE,
  heatmap_legend_param = list(title = 'Mean difference'),
  column_title = 'Cipro - Control'
)

combo <- main_ht + delta_ht
highlight_legend <- Legend(
  title = 'Highlight',
  labels = 'Row variance  90th pct',
  legend_gp = gpar(fill = '#ffe0b2', col = '#ff9500', lwd = 1.2)
)
combo <- draw(
  combo,
  heatmap_legend_side = 'right',
  annotation_legend_side = 'right',
  heatmap_legend_list = list(highlight_legend)
)
```



7. Notes

```
cat('Highlighted rows:', length(highlight_rows), '\n')
```

```
## Highlighted rows: 8
```

```
cat('Top genomes in highlight:\n')
```

```
## Top genomes in highlight:
```

```
print(sort(table(row_genome[highlight_rows]), decreasing = TRUE))
```

```
## Turicimonas_muris_YL45
##                      8
```

The decoration makes it obvious that the most variable SNPs mostly belong to the Akkermansia genome in this subset, and the companion Δ-mean heatmap shows where Ciprofloxacin drives allele-frequency changes relative to Control.