Day 2 – 02 Simple Heatmap (Exercises – Solutions)

Seminar practice worksheet

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

```r
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))
## ========================================
```

```r
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))
## ========================================
```

```r
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

```r
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
```

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

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

```r
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
```

```r
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

```r
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_I148
##                            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_I148    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]
```

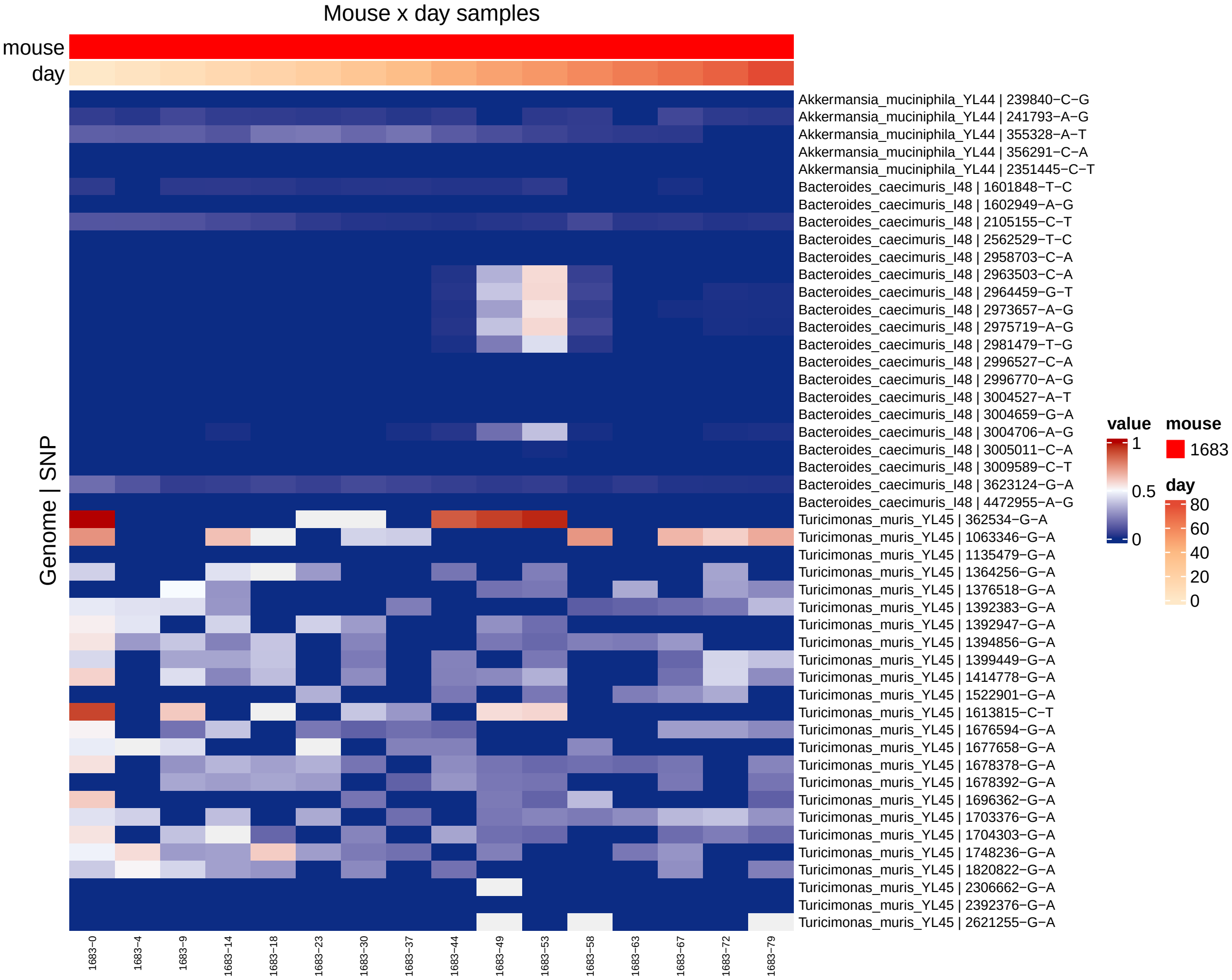
6. Colors and annotations

```
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
```

```r
if (abs(max_val - min_val) < .Machine$double.eps) {
  max_val <- min_val + 1
}
mid_val <- (min_val + max_val) / 2
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')
```



```r
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
```

```r
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.