Day 2 – 04 Full Heatmap (Exercises)

Seminar practice worksheet

This practice notebook guides you through rebuilding the "all genomes" heatmap (`dataset3_subset*.csv`). Compared with earlier exercises, you now juggle a larger matrix, multiple annotation layers, and custom ordering. Follow the prompts, fill in each `# TODO`, and keep notes on the design decisions you make.

> **Tip:** Run `scripts/00_prepare.Rmd` first so `ComplexHeatmap`, `circlize`, and helper packages are ready. If the dataset is missing, regenerate it via `data/00_prepare_dataset.Rmd`.

1. Packages, paths, helper settings

Set yourself up for success by loading packages, pointing to the dataset files, and defining constants you will re-use (e.g., the NA color and PDF output path). Add any palettes you plan to try so they are available later.

```
# TODO: library(ComplexHeatmap); library(circlize); library(viridisLite)
# TODO: 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')
# TODO: na_color <- '#dcdcdc'
# TODO: set.seed(...) if you plan to shuffle or sample
```

Reflection: Why do we keep both the wide and long versions even though the heatmap consumes only the wide table?

2. Load data and report missingness

Read both tables, print their dimensions, and log the total number of NA values in the `value` column. If NAs are present, produce a matrix (mouse vs day) that shows where they occur so you can make informed filtering choices later.

```
# TODO: wide_df <- read.csv(subset_path, check.names = FALSE, stringsAsFactors = FALSE)
# TODO: long_df <- read.csv(long_path, check.names = FALSE, stringsAsFactors = FALSE)
# TODO: cat('Wide rows x cols:', nrow(wide_df), ncol(wide_df), '\n')
# TODO: cat('Long rows x cols:', nrow(long_df), ncol(long_df), '\n')
# TODO: na_total <- sum(is.na(long_df$value))
# TODO: if (na_total > 0) { build tapply(...) to locate them }
```

Question: Does the larger dataset introduce NA clusters concentrated in a single mouse/treatment?

3. Build the numeric matrix

Construct the heatmap matrix using the sample columns, ensuring you keep row labels informative for debugging. Reuse or rebuild a `sample_meta` data frame that captures `mouse_id`, `day`, `treatment_group`, and `sample_id`. This metadata will power your annotations and ordering.

```
# TODO: sample_cols <- setdiff(names(wide_df), c('Genome','snp_id','Position'))
# TODO: mat <- as.matrix(wide_df[, sample_cols]); mode(mat) <- 'numeric'
# TODO: rownames(mat) <- paste(wide_df$Genome, wide_df$snp_id, sep = ' | ')
# TODO: sample_meta <- unique(long_df[, c('mouse_id','day','treatment_group')])
# TODO: sample_meta$sample_id <- paste(sample_meta$mouse_id, sample_meta$day, sep='-')
# TODO: sample_meta <- sample_meta[match(colnames(mat), sample_meta$sample_id), ]
```

Check: Use `stopifnot(identical(colnames(mat), sample_meta$sample_id))` to catch mismatches early.

4. Baseline heatmap and color experiments

Start with a minimal `Heatmap` (no clustering, no annotations) so you can verify the matrix orientation. Next, experiment with at least two color palettes by constructing separate `colorRamp2` functions. Document which palette you prefer and why (contrast, perceptual ordering, etc.).

```
# TODO: mins <- min(mat, na.rm = TRUE); maxs <- max(mat, na.rm = TRUE); mids <- (mins+maxs)/2
# TODO: palette_a <- circlize::colorRamp2(c(mins, mids, maxs), c('#0c2c84','#f7fbff','#b30000'))
# TODO: palette_b <- circlize::colorRamp2(c(mins, mids, maxs), viridisLite::viridis(3))
# TODO: Heatmap(mat, name='value', col=palette_a, na_col = na_color)
# TODO: Heatmap(mat, name='value', col=palette_b, na_col = na_color)
```

Reflection: Which palette makes low values easiest to distinguish once annotations are added?

5. Column annotations and ordering

Build annotations that explain treatment group and baseline/post-antibiotic status. Then define an ordering (e.g., order by `mouse_id`, then `day`) and apply it consistently to both the matrix and the annotation object. Draw the heatmap again to confirm columns appear in the intended sequence.

```
# TODO: sample_meta$post_ab <- ifelse(sample_meta$day == 0, 'baseline', 'post')
# TODO: col_ann <- HeatmapAnnotation(
#         treatment = sample_meta$treatment_group,
#         status = sample_meta$post_ab,
#         annotation_name_side = 'left'
#       )
# TODO: order_idx <- order(sample_meta$mouse_id, sample_meta$day)
# TODO: mat_ordered <- mat[, order_idx]
# TODO: col_ann_ordered <- col_ann[order_idx]
# TODO: Heatmap(mat_ordered, name='value', top_annotation = col_ann_ordered,
#               cluster_rows = FALSE, cluster_columns = FALSE, na_col = na_color)
```

Check: Are replicates or treatment switches now easier to spot?

6. Annotation enhancements, column splits, and readable row titles

Recreate the richer annotation stack from `03_heatmap_annotations.Rmd`, but now apply it to the full dataset. Ideas to try:

- Encode `treatment_group` with a named color vector so the legend matches your slide deck.
- Add a continuous day gradient using `anno_simple()` so viewers can track the time axis directly in the annotation bar.
- Overlay mouse IDs (rotated text or a thin color strip) to highlight replicate structure.
- Optionally, split rows by genome (`row_split`) to mimic the multi-panel layout from the guided notebook, and either shorten those labels or shrink the font so they do not overlap in the PDF.
- Add a `column_split` (e.g., by treatment group) so the plot is chunked into digestible vertical blocks in addition to the chronological ordering.

```
# TODO: treatment_cols <- c(...); status_cols <- c(...)
# TODO: day_col_fun <- circlize::colorRamp2(range(sample_meta$day), c('#f7fbff','#084594'))
# TODO: col_ann_rich <- HeatmapAnnotation(
#         treatment = anno_simple(sample_meta$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$mouse_id, rot = 90, gp = grid::gpar(fontsize = 6)),
#         annotation_name_side = 'left'
#       )
# TODO: col_ann_rich <- col_ann_rich[order_idx]
# TODO: column_split <- factor(sample_meta$treatment_group[order_idx], levels = treatment_levels)
# TODO: row_split <- factor(wide_df$Genome, levels = unique(wide_df$Genome))
# TODO: row_titles <- gsub('_.*', '', levels(row_split)) # or set fontsize via row_title_gp
# TODO: Heatmap(mat_ordered, name='value', col = palette_a, top_annotation = col_ann_rich,
#               cluster_rows = TRUE, row_split = row_split, column_split = column_split,
#               na_col = na_color, show_row_names = FALSE, show_column_names = FALSE,
#               row_title = row_titles, row_title_gp = grid::gpar(fontsize = 9))
```

Reflection: Which annotation layer (treatment colors, day gradient, mouse IDs) helped the most when interpreting the plot? Keep notes for your presentation.

7. Row filtering via variance

Large matrices can hide structure. Compute per-row variance, keep the top 100 rows (or another threshold), and draw a diagnostic heatmap to see whether the higher-variance SNPs highlight patterns that were previously buried. Remember to reapply the column ordering and annotations.

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

Reflection: Does variance filtering change the biological story you would tell?

8. Final polish and PDF export

Combine your favorite palette, annotations, column order, and a row clustering strategy into a final heatmap. Add a column title and tweak legend names if needed. Export the final figure to `pdf_path` so it can be shared outside the notebook.

```
# TODO: ht_final <- Heatmap(
#         mat_ordered,
#         name = 'value',
#         col = palette_a,
#         top_annotation = col_ann_rich,
#         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)'
#       )
# TODO: draw(ht_final)
# TODO: pdf(pdf_path, width = 11, height = 7); draw(ht_final); dev.off();
#       cat('Saved heatmap to', pdf_path, '\n')
```

Question: Which design choices would you highlight if you presented this plot in lab meeting (ordering rationale, annotation selection, etc.)?

9. Notes and extensions

Use this space to capture observations, alternative palettes, or next steps (e.g., adding row annotations, trying `row_split`, exporting PNGs).

```
# TODO: jot down thoughts, ideas, or additional code experiments
```

Once you're happy with the exercise output, compare it against the solution notebook (`scripts/04_full_heatmap_exercises_`
to fill any gaps.