Day 2 – 02 Simple Heatmap

Seminar plotting walk-through

A minimal example that turns the prepared wide table (`dataset1_subset.csv`) into a heatmap with practically no extra formatting. This is the first step before moving on to the fully annotated version in `03_heatmap_annotations.Rmd`.

1. Packages and paths

```
library(ComplexHeatmap)
library(circlize)
library(viridisLite)
subset_path <- file.path('..', 'data', 'dataset1_subset.csv')
long_path <- file.path('..', 'data', 'dataset1_subset_long.csv')
pdf_path <- file.path('..', 'pdf', 'dataset1_heatmap_basic.pdf')
target_group <- 'Control'  # change this to view other treatment groups
na_color <- '#dcdcdc'
```

2. Load, subset by treatment group, and convert to matrices

```
wide_df <- read.csv(subset_path, check.names = FALSE, stringsAsFactors = FALSE)
long_df <- read.csv(long_path, check.names = FALSE, stringsAsFactors = FALSE)
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_cols <- setdiff(names(wide_df), c('Genome', 'snp_id', 'Position'))

mat_all <- as.matrix(wide_df[, sample_cols])
mode(mat_all) <- 'numeric'
rownames(mat_all) <- paste(wide_df$Genome, wide_df$snp_id, sep = ' | ')
colnames(mat_all) <- sample_cols

keep_cols <- intersect(sample_cols, sample_meta$sample_id[sample_meta$treatment_group == target_group])
if (!length(keep_cols)) {
  stop('No samples found for treatment group: ', target_group)
}
heatmap_matrix <- mat_all[, keep_cols, drop = FALSE]
colnames(heatmap_matrix) <- keep_cols
```

3. Choose a data-aware color scale

Allele frequencies naturally lie between 0 and 1. Using a diverging palette with an assumed midpoint (e.g., 0.5) suggests there is a special reference level, which is misleading for these measurements. Instead, derive a sequential palette from the observed range so color intensity increases smoothly with the allele frequency.

```
value_range <- range(heatmap_matrix, na.rm = TRUE)
color_breaks <- seq(value_range[1], value_range[2], length.out = 5)
```

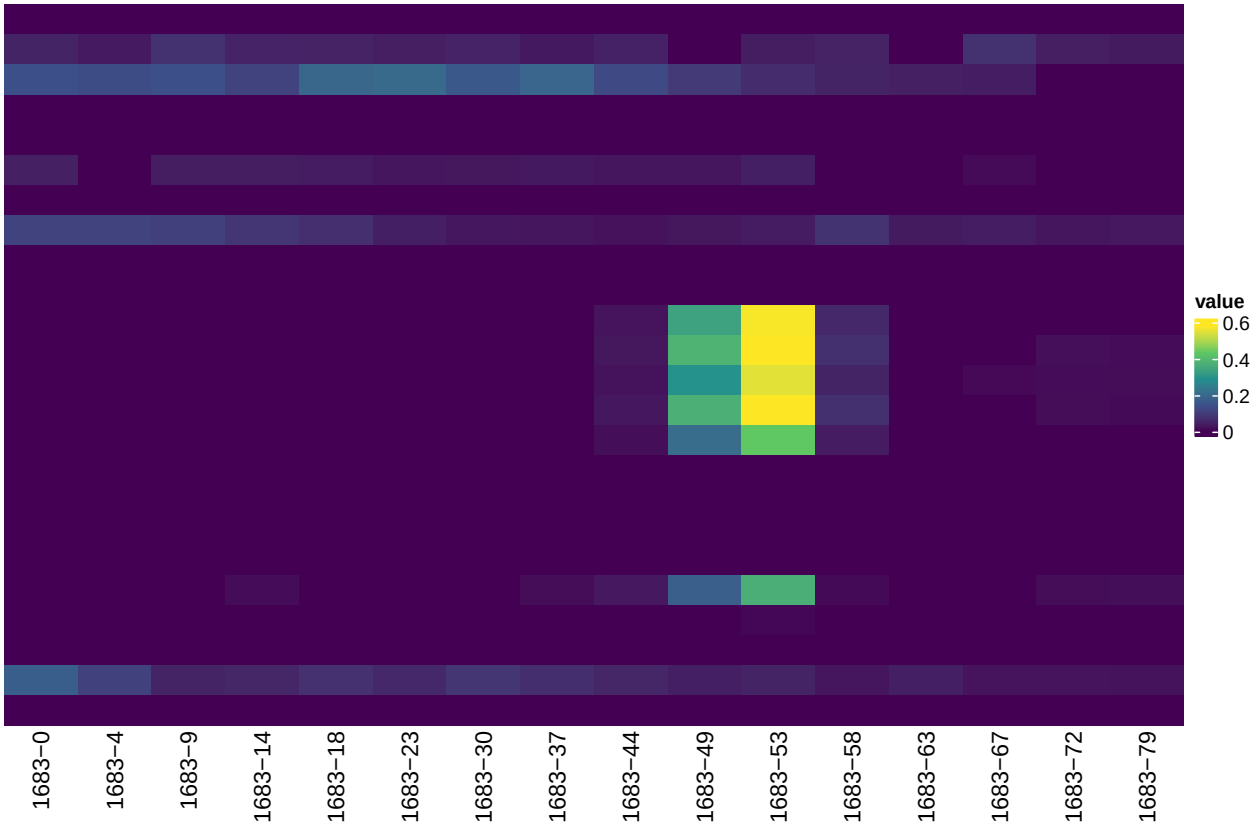
```r
color_fun <- circlize::colorRamp2(color_breaks, viridisLite::viridis(5))
```

Feel free to swap in other sequential palettes (e.g., `RColorBrewer`) as long as the hues progress monotonically with the data.

4. Draw the simplest possible heatmap

```r
ht <- Heatmap(
  heatmap_matrix,
  name = 'value',
  col = color_fun,
  cluster_rows = FALSE,
  cluster_columns = FALSE,
  show_row_names = FALSE,
  show_column_names = TRUE,
  na_col = na_color
)

draw(ht)
```



5. Optional: highlight treatment and baseline/post status

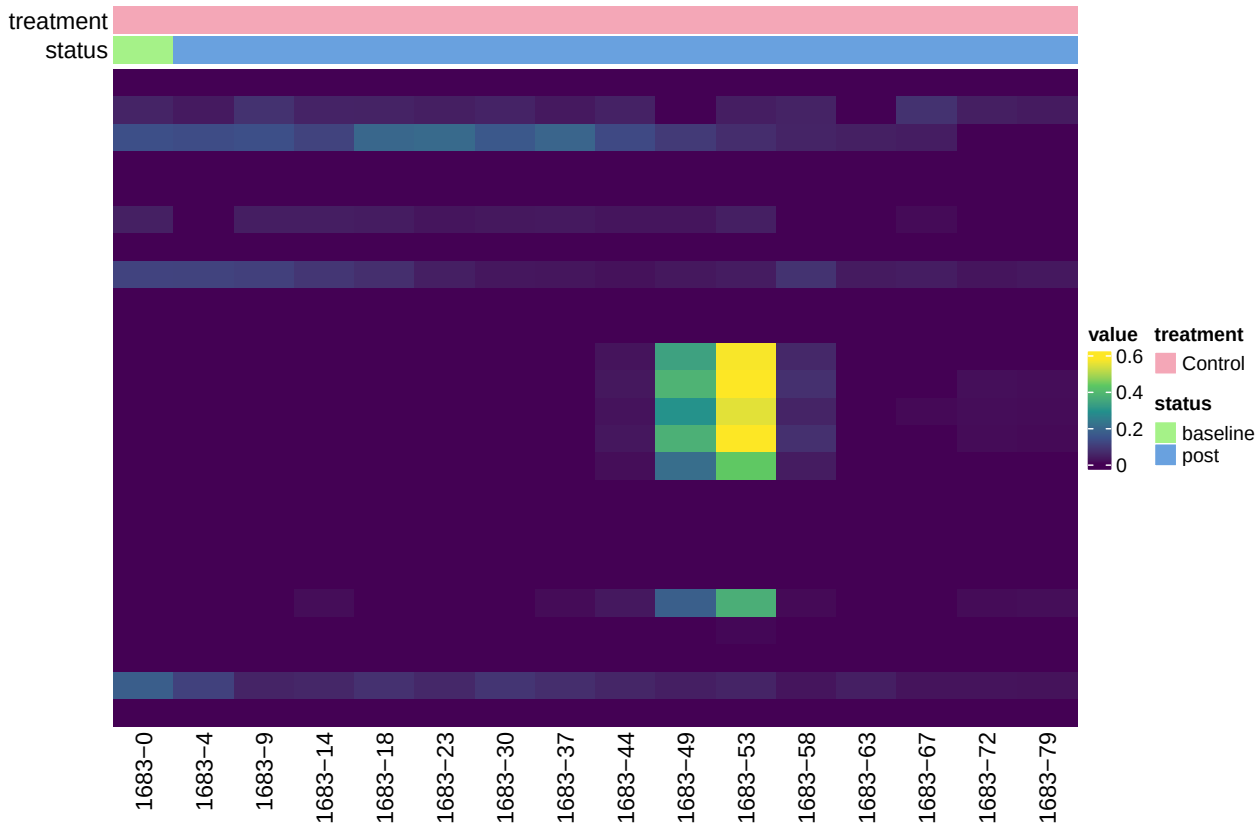
Before moving on, you can add a light-weight column annotation labeling each sample with its treatment group and whether it was collected at baseline (day 0) or post-antibiotic (day > 0).

```r
sample_info <- sample_meta[sample_meta$sample_id %in% colnames(heatmap_matrix), ]
sample_info <- sample_info[match(colnames(heatmap_matrix), sample_info$sample_id), ]
sample_info$post_ab <- ifelse(sample_info$day == 0, 'baseline', 'post')
```

```
col_ann <- HeatmapAnnotation(
  treatment = sample_info$treatment_group,
  status = sample_info$post_ab,
  annotation_name_side = 'left'
)

ht_ann <- Heatmap(
  heatmap_matrix,
  name = 'value',
  col = color_fun,
  cluster_rows = FALSE,
  cluster_columns = FALSE,
  show_row_names = FALSE,
  show_column_names = TRUE,
  na_col = na_color,
  top_annotation = col_ann
)

draw(ht_ann)
```



6. Optional: compare two treatment groups side-by-side

If you want to compare two treatment groups visually, filter the full matrix to those samples and split the columns by treatment. This keeps the code simple while showing how cohorts differ.

```
compare_groups <- c('Control', 'Ciprofloxacin')
keep_compare <- sample_meta$sample_id[sample_meta$treatment_group %in% compare_groups]
```

```
compare_mat <- mat_all[, keep_compare, drop = FALSE]
group_factor <- factor(sample_meta$treatment_group[match(colnames(compare_mat), sample_meta$sample_id)]
                       levels = compare_groups)
Heatmap(compare_mat,
        name = 'value',
        col = color_fun,
        column_split = group_factor,
        cluster_rows = FALSE,
        cluster_columns = FALSE,
        show_row_names = FALSE,
        na_col = na_color) |>
  draw()
```

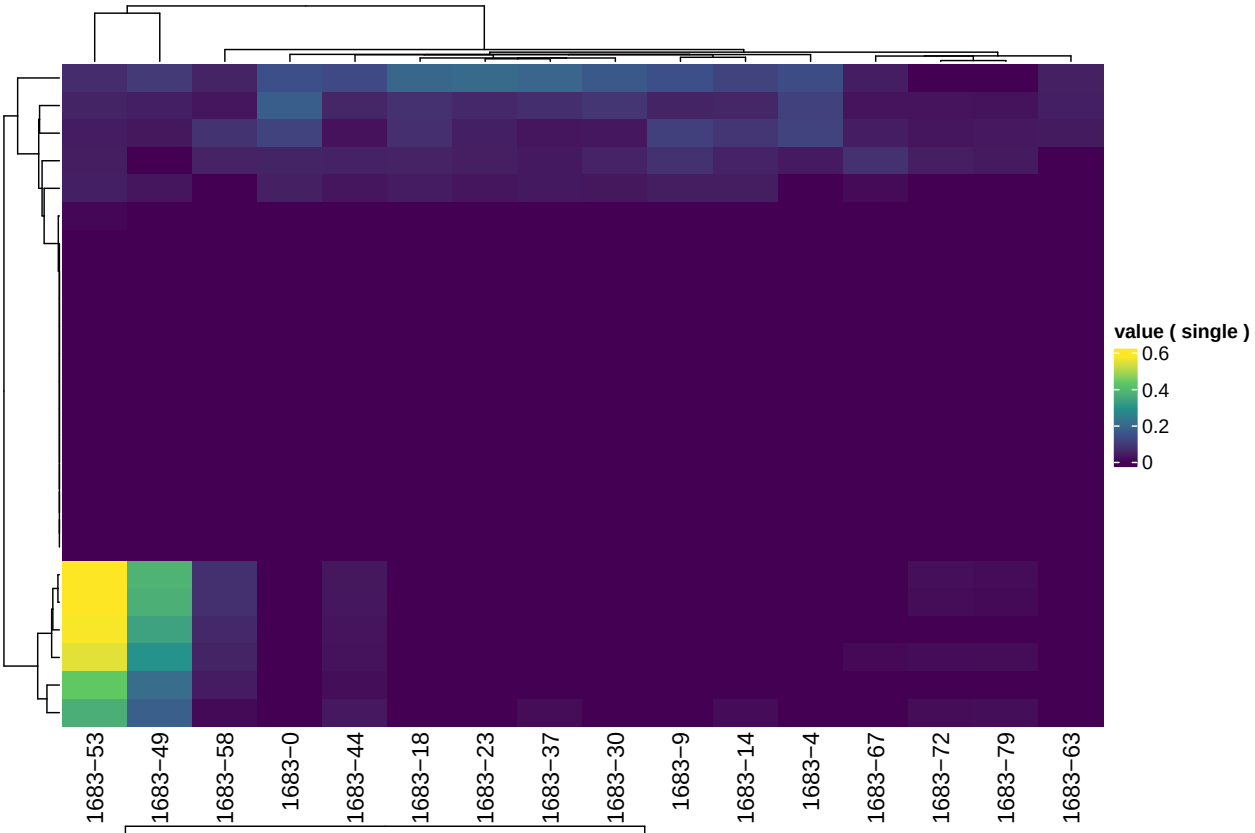
7. Optional: save to PDF

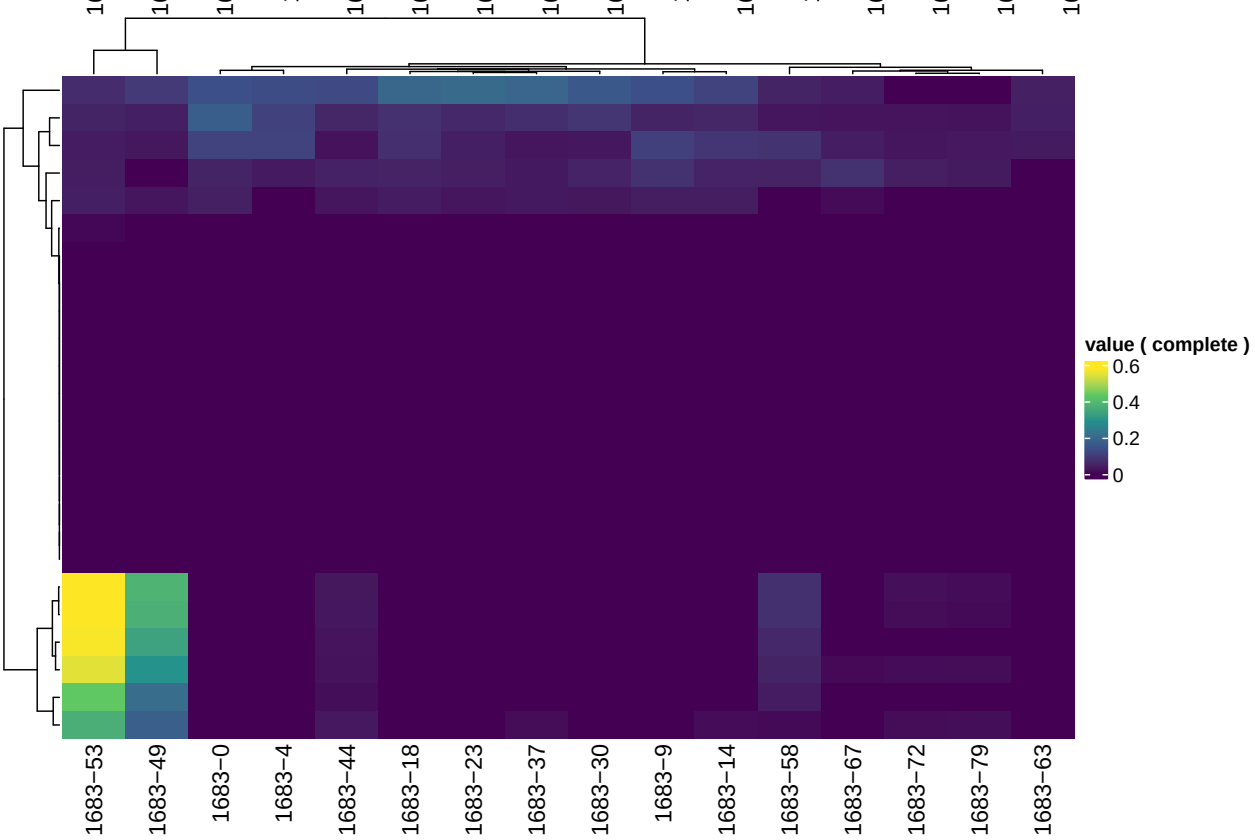
```
pdf(pdf_path, width = 10, height = 7)
draw(ht)
dev.off()
cat('Saved basic heatmap to', pdf_path, '\n')
```

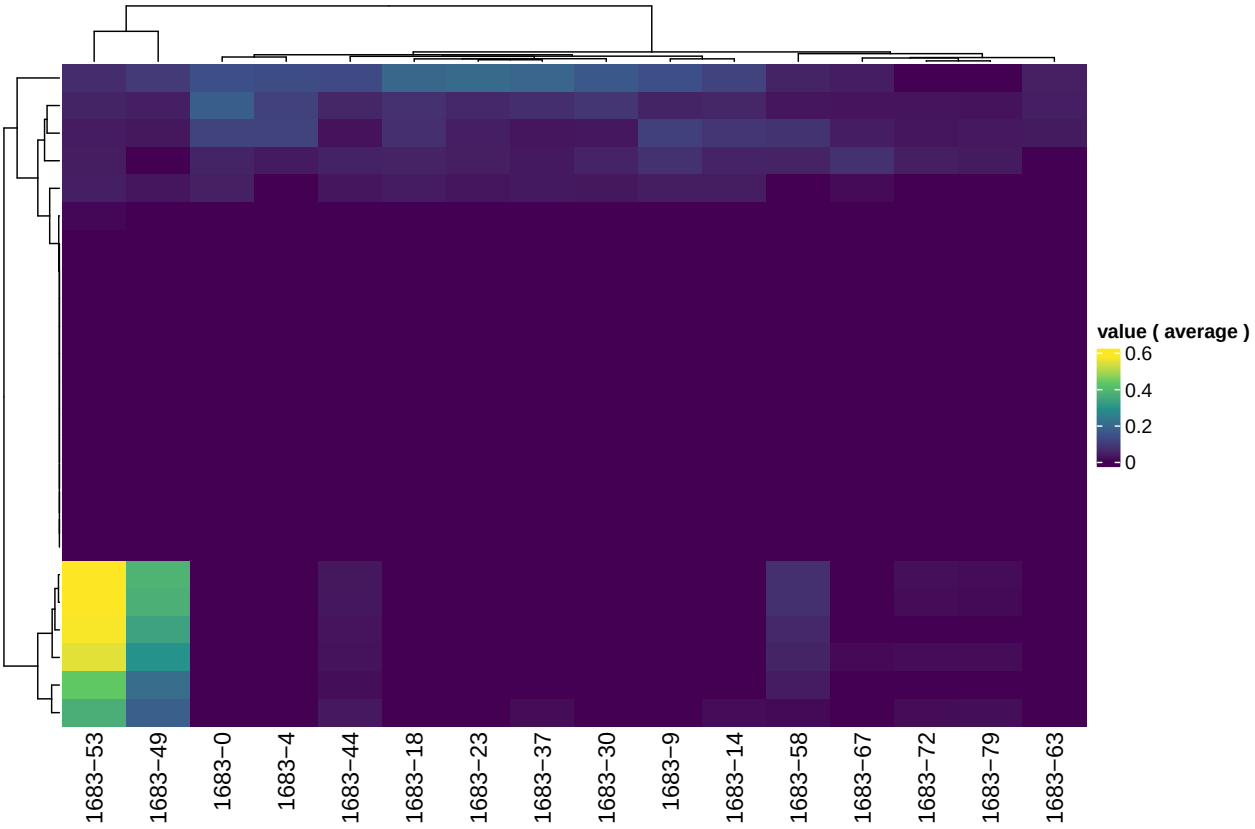
8. Try a few clustering methods

Even though this notebook is meant to stay simple, it is still useful to peek at what happens when we let ComplexHeatmap cluster rows/columns. Below are three variants that only differ in the linkage method.

```
methods <- c('single', 'complete', 'average')
for (m in methods) {
  ht_clust <- Heatmap(
    heatmap_matrix,
    name = paste('value (', m, ')'),
    col = color_fun,
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    clustering_method_rows = m,
    clustering_method_columns = m,
    show_row_names = FALSE,
    show_column_names = TRUE,
    na_col = na_color
  )
  grid::grid.newpage()
  draw(ht_clust, main_heatmap = 1)
}
```


value (single)
0.6
0.4
0.2
0
1683−53
1683−49
1683−58
1683−0
1683−44
1683−18
1683−23
1683−37
1683−30
1683−9
1683−14
1683−4
1683−67
1683−72
1683−79
1683−63


value (complete)
0.6
0.4
0.2
0
1683−53
1683−49
1683−0
1683−4
1683−44
1683−18
1683−23
1683−37
1683−30
1683−9
1683−14
1683−58
1683−67
1683−72
1683−79
1683−63

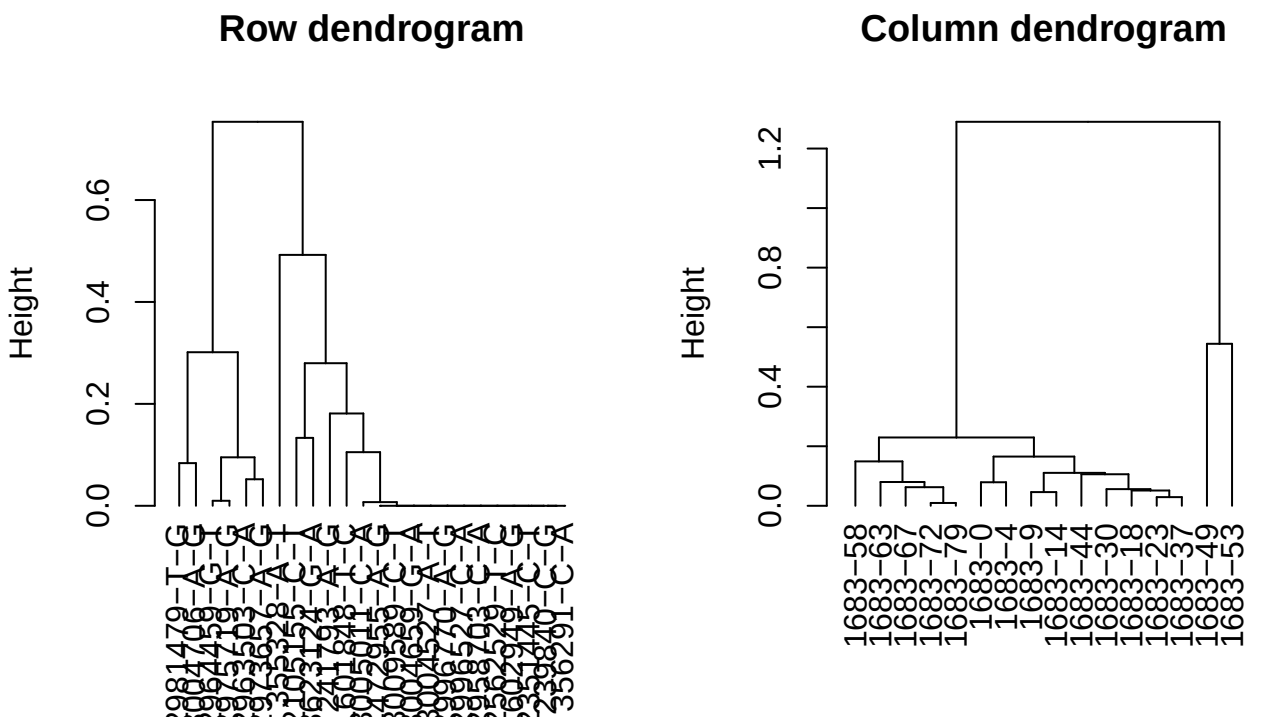


9. Optional: inspect dendrograms directly

Sometimes you just want to see the clustering tree without the heatmap. You can reuse the ordered matrix to build a dendrogram using base R functions:

```
row_dend <- as.dendrogram(hclust(dist(heatmap_matrix), method = 'complete'))
col_dend <- as.dendrogram(hclust(dist(t(heatmap_matrix)), method = 'complete'))

par(mfrow = c(1, 2))
plot(row_dend, main = 'Row dendrogram', ylab = 'Height')
plot(col_dend, main = 'Column dendrogram', ylab = 'Height')
```

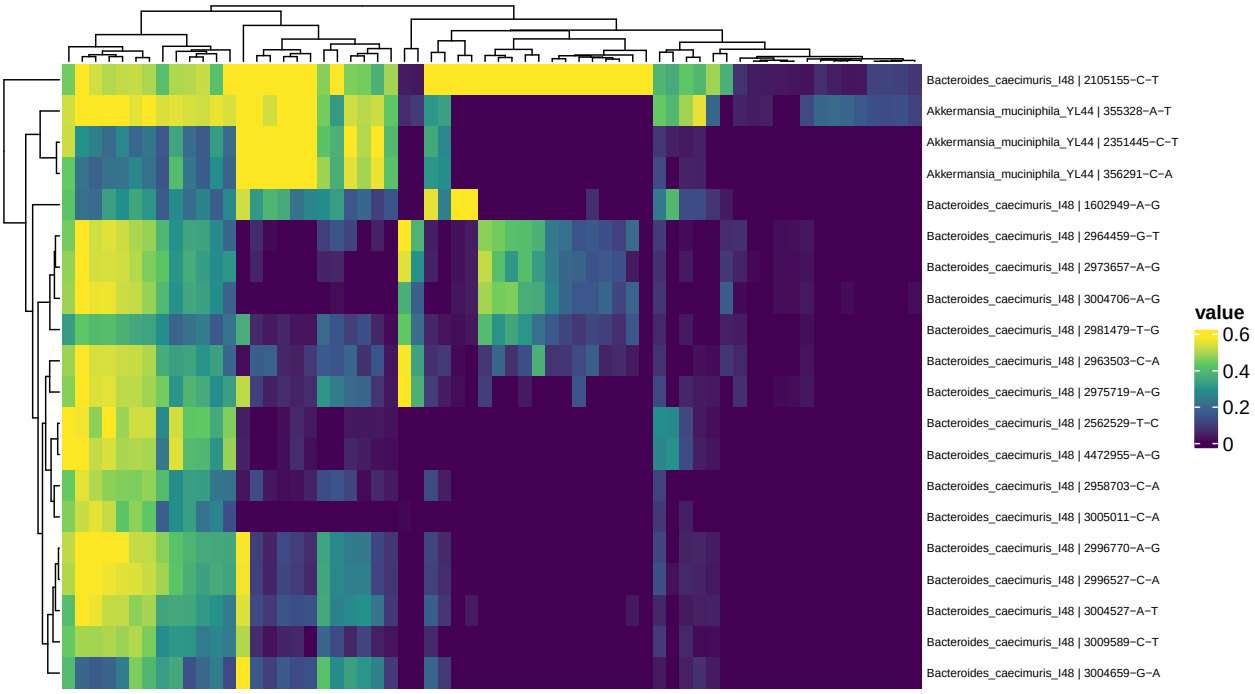



```
par(mfrow = c(1, 1))
```

10. Which SNPs vary the most?

To focus on the most dynamic rows, compute the variance of each SNP and plot a small heatmap of the top 20. This helps students see where the biggest changes occur before moving to the advanced notebook.

```
row_var <- apply(mat_all, 1, var, na.rm = TRUE)
top_idx <- order(row_var, decreasing = TRUE)[seq_len(min(20, length(row_var)))]
ht_top <- Heatmap(
  mat_all[top_idx, , drop = FALSE],
  name = 'value',
  col = color_fun,
  show_row_names = TRUE,
  row_names_gp = grid::gpar(fontsize = 6),
  show_column_names = FALSE,
  na_col = na_color
)
draw(ht_top)
```



Ready for more control? Open `03_heatmap_annotations.Rmd` to add ordering, annotations, and publication polish.