

Clinical heatmap visualization

MAXOMOD_CSF

2026-02-16

Contents

Overview	1
Setup	2
Load packages	2
Project root and parameters	2
Load and validate input	2
Prepare data and annotations	3
Annotation colors	4
Top 100 variable proteins and scaling	5
Save heatmaps	5
EPS	5
PDF	6
Preview in HTML	7
Outputs	7

Overview

This document reproduces the **clinical heatmap** step from the pipeline. It reads a normalized and imputed `SummarizedExperiment` object from **demo/Discovery/02_Missing_Inspection/** and saves figures into the **Plots** folder. Ensure the demo folder has been populated (e.g. copy from **Discovery/02_Missing_Inspection/**) before running.

Equivalent command line:

```
Rscript 07_Clinical_heatmap.R --input  
↪ demo/Discovery/02_Missing_Inspection/norm_imp_MinProb.rds --output Plots --seed 9
```

To run with different inputs, change the `params` in the YAML header above and re-knit, or use **Knit** → **Knit with Parameters** in RStudio.

Setup

Load packages

```
library(pheatmap)
library(circlize)
library(ComplexHeatmap)
library(RColorBrewer)
library(SummarizedExperiment)
library(DEP)
```

Project root and parameters

Paths are relative to the **project root** (MAXOMOD_CSF). When the Rmd is in `vignettes/`, we detect that and resolve all paths from the project root so the document works regardless of current working directory. **Run this chunk first** when running chunks interactively.

```
# Project root: one level up from vignettes/, or current directory
project_root <- if (basename(getwd()) == "vignettes") {
  normalizePath("../", winslash = "/")
} else {
  getwd()
}
setwd(project_root)
message("Project root: ", project_root)
```

```
## Project root: /Users/xliu2942/Documents/Projects/MAXOMOD/MAXOMOD_CSF
```

```
# From Rmd params (edit in YAML or use "Knit with Parameters")
input_path      <- params$input
input_path_full <- file.path(project_root, params$input)
output_dir      <- file.path(project_root, params$output)
set.seed(params$seed)
```

Load and validate input

```

if (is.null(input_path) || input_path == "") {
  stop("Please set 'input' in the document parameters (YAML or Knit with Parameters).")
}
if (!file.exists(input_path_full)) {
  stop("Input file does not exist: ", input_path_full, ". Run the pipeline to generate it
  ↪ or copy it into demo/Discovery/02_Missing_Inspection/.")
}

input <- readRDS(input_path_full)

```

```

if (is.null(output_dir) || output_dir == "") {
  output_dir <- getwd()
  message("No output path supplied; using current directory.")
} else {
  if (!dir.exists(output_dir)) {
    dir.create(output_dir, recursive = TRUE)
  }
}

```

Prepare data and annotations

```

expression <- assay(input)
metadata    <- input@colData

# Order samples: ctrl first, als last
ordered_samples <- rownames(metadata)[order(metadata$condition == "als")]
mat             <- expression[, ordered_samples]

# Column annotation (clinical variables)
anno_col <- as.data.frame(metadata[ordered_samples, c(
  "condition",
  "progression_group",
  "progression_rate",
  "slow_vital_capacity",
  "pNFh",
  "Nf1",
  "sex",
  "age",
  "age_at_onset",
  "genetics",
  "onset",
  "ECAS"
)])

# Treat "not_performed" in genetics as missing
anno_col$genetics[anno_col$genetics == "not_performed"] <- NA

stopifnot(identical(rownames(anno_col), colnames(mat)))

```

Annotation colors

```
ha_colors <- list(
  condition = c(ctrl = "black", als = "#D73027"),
  sex = c(Female = "#E78AC3", Male = "#66C2A5"),
  Nfl = colorRamp2(
    quantile(anno_col$Nfl, probs = c(0, 0.5, 1), na.rm = TRUE),
    c("white", "orchid", "purple4")
  ),
  pNFh = colorRamp2(
    quantile(anno_col$pNFh, probs = c(0, 0.5, 1), na.rm = TRUE),
    c("white", "orchid", "purple4")
  ),
  progression_rate = colorRamp2(
    quantile(anno_col$progression_rate, probs = c(0, 0.5, 1), na.rm = TRUE),
    c("white", "orangered", "darkred")
  ),
  slow_vital_capacity = colorRamp2(
    quantile(anno_col$slow_vital_capacity, probs = c(0, 0.5, 1), na.rm = TRUE),
    c("white", "orangered", "darkred")
  ),
  age = colorRamp2(
    quantile(anno_col$age, probs = c(0, 0.5, 1), na.rm = TRUE),
    c("white", "yellowgreen", "darkgreen")
  ),
  age_at_onset = colorRamp2(
    quantile(anno_col$age_at_onset, probs = c(0, 0.5, 1), na.rm = TRUE),
    c("white", "yellowgreen", "darkgreen")
  ),
  progression_group = c(SP = "#1B9E77", IP = "#D95F02", FP = "#7570B3"),
  onset = c(spinal = "#FDAE61", bulbar = "#FEE090"),
  genetics = c(
    negative = "grey",
    C9orf72 = "firebrick",
    VUS = "orange",
    ROCK = "darkgreen",
    SOD1 = "skyblue",
    SOD1_FIG4 = "darkviolet"
  ),
  ECAS = colorRamp2(
    quantile(anno_col$ECAS, probs = c(0, 0.5, 1), na.rm = TRUE),
    c("lightblue", "deepskyblue", "darkblue")
  )
)

ha <- HeatmapAnnotation(
  df = anno_col,
  col = ha_colors,
  annotation_height = unit(4, "mm"),
  na_col = "white",
  annotation_name_side = "left"
)
```

Top 100 variable proteins and scaling

```
row_vars <- apply(mat, 1, var, na.rm = TRUE)
top100 <- names(sort(row_vars, decreasing = TRUE))[1:100]
mat <- mat[top100, ]

scaled_mat <- t(scale(t(mat)))
scaled_mat[scaled_mat > 3] <- 3
scaled_mat[scaled_mat < -3] <- -3

col_fun <- colorRamp2(c(-3, 0, 3), c("#1B5B9D", "white", "#D7191C"))
```

Save heatmaps

EPS

```
cairo_ps(
  file = file.path(output_dir, "clinical_heatmap.eps"),
  width = 16/1.6, height = 14/2, onefile = FALSE, bg = "transparent"
)
ComplexHeatmap::draw(ComplexHeatmap::Heatmap(
  scaled_mat,
  na_col = "white",
  col = col_fun,
  cluster_rows = TRUE,
  cluster_columns = FALSE,
  show_row_names = FALSE,
  show_column_names = FALSE,
  row_dend_width = unit(0, "mm"),
  show_row_dend = FALSE,
  top_annotation = ha,
  heatmap_legend_param = list(
    at = c(-3, -2, -1, 0, 1, 2, 3),
    color_bar = "continuous",
    legend_height = unit(4, "cm"),
    title = NULL
  )
), heatmap_legend_side = "left",
  annotation_legend_side = "right",
  padding = unit(c(5, 28, 5, 2), "mm"))
dev.off()
```

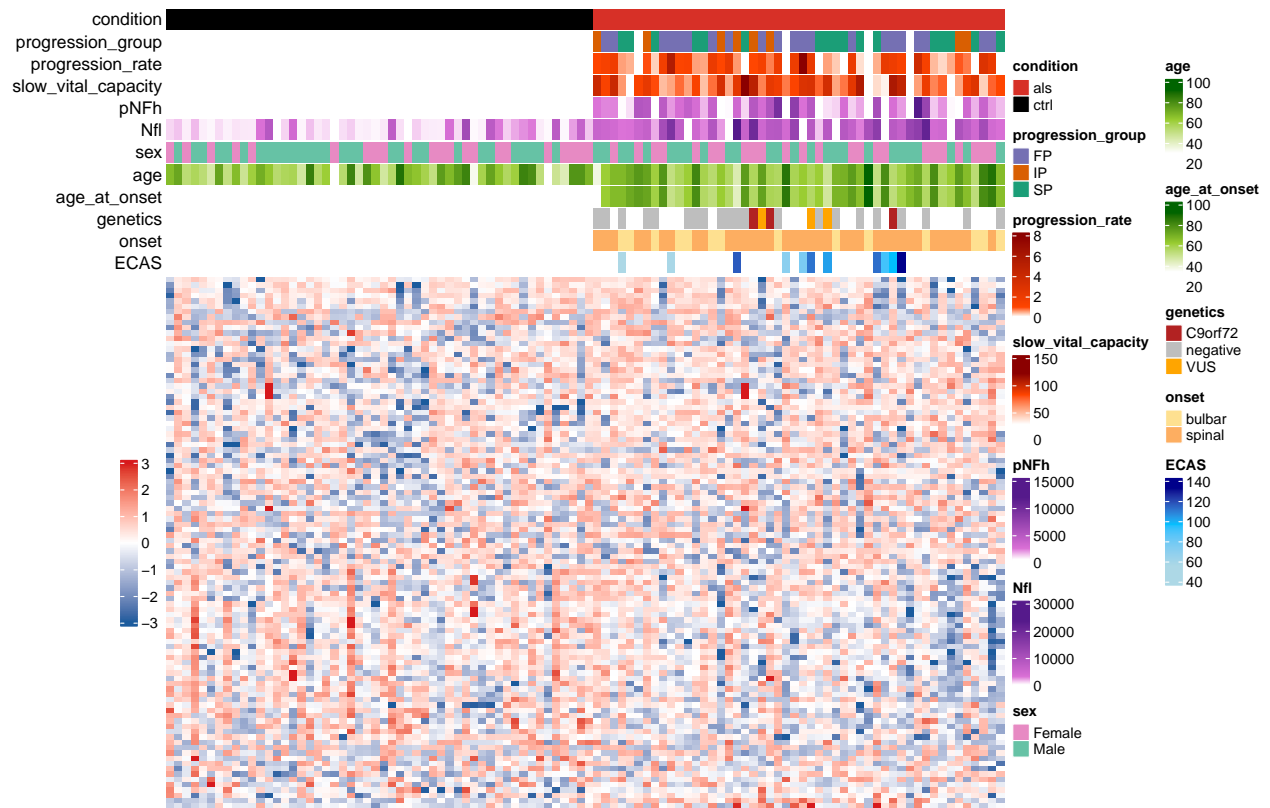
```
## pdf
## 2
```

PDF

```
pdf(  
  file = file.path(output_dir, "clinical_heatmap.pdf"),  
  width = 16/1.6, height = 14/2, onefile = FALSE, bg = "transparent"  
)  
ComplexHeatmap::draw(ComplexHeatmap::Heatmap(  
  scaled_mat,  
  na_col = "white",  
  col = col_fun,  
  cluster_rows = TRUE,  
  cluster_columns = FALSE,  
  show_row_names = FALSE,  
  show_column_names = FALSE,  
  row_dend_width = unit(0, "mm"),  
  show_row_dend = FALSE,  
  top_annotation = ha,  
  heatmap_legend_param = list(  
    at = c(-3, -2, -1, 0, 1, 2, 3),  
    color_bar = "continuous",  
    legend_height = unit(4, "cm"),  
    title = NULL  
  )  
) , heatmap_legend_side = "left",  
  annotation_legend_side = "right",  
  padding = unit(c(5, 28, 5, 2), "mm"))  
dev.off()
```

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

Preview in HTML



Outputs

- **EPS:** /Users/xliu2942/Documents/Projects/MAXOMOD/MAXOMOD_CSF/Plots/Fig1b_clinical_heatmap/clinical
- **PDF:** /Users/xliu2942/Documents/Projects/MAXOMOD/MAXOMOD_CSF/Plots/Fig1b_clinical_heatmap/clinical

Input is read from **demo/Discovery/02_Missing_Inspection/** (subfolder name unchanged). Figures are saved in the **Plots** folder. Run from the project root.