

Day 2 – 00 Prepare Dataset

Seminar plotting walk-through

This notebook converts the raw Day 2 file (`second_day_part2/data/dataset1.csv`) into the tidy CSVs used throughout the course:

- `second_day_part2/data/dataset1_subset.csv` / `dataset1_subset_long.csv`: only the two spot-light genomes (Akkermansia + Bacteroides).
- `second_day_part2/data/dataset2_subset.csv` / `dataset2_subset_long.csv`: the same table but with a third genome (Turicimonas) added for the exercises track.

Everything is written with base R so that you can explain each step to beginners. Run it from the repository root:

```
R -e "rmarkdown::render('second_day_part2/data/00_prepare_dataset.Rmd')"
```

1. Define input/output paths

```
input_path <- 'dataset1.csv'
subset_path <- 'dataset1_subset.csv'
long_path <- 'dataset1_subset_long.csv'
subset2_path <- 'dataset2_subset.csv'
long2_path <- 'dataset2_subset_long.csv'
group_map_path <- 'mouse_group_mapping.csv'
```

2. Load the raw CSV and inspect

```
raw_df <- read.csv(
  input_path,
  check.names = FALSE,
  stringsAsFactors = FALSE,
  fileEncoding = 'UTF-8-BOM'
)

cat('Rows:', nrow(raw_df), '\\nColumns:', ncol(raw_df), '\\n')
```

```
## Rows: 71 \nColumns: 69 \n
```

```
colnames(raw_df)
```

##	[1]	"Genome"	"Position"	"Alternative"	"Reference"	"Feature"
##	[6]	"1683-0"	"1683-4"	"1683-9"	"1683-14"	"1683-18"
##	[11]	"1683-23"	"1683-30"	"1683-37"	"1683-44"	"1683-49"
##	[16]	"1683-53"	"1683-58"	"1683-63"	"1683-67"	"1683-72"
##	[21]	"1683-79"	"1688-0"	"1688-4"	"1688-9"	"1688-14"
##	[26]	"1688-18"	"1688-23"	"1688-30"	"1688-37"	"1688-44"
##	[31]	"1688-49"	"1688-53"	"1688-58"	"1688-63"	"1688-67"
##	[36]	"1688-72"	"1688-79"	"1692-0"	"1692-4"	"1692-9"
##	[41]	"1692-14"	"1692-18"	"1692-23"	"1692-30"	"1692-37"

```
## [46] "1692-44"      "1692-49"      "1692-53"      "1692-58"      "1692-63"
## [51] "1692-67"      "1692-72"      "1692-79"      "1699-0"       "1699-4"
## [56] "1699-9"       "1699-14"      "1699-18"      "1699-23"      "1699-30"
## [61] "1699-37"      "1699-44"      "1699-49"      "1699-53"      "1699-58"
## [66] "1699-63"      "1699-67"      "1699-72"      "1699-79"
```

```
head(raw_df[, 1:8])
```

```
##              Genome Position Alternative Reference
## 1 Akkermansia_muciniphila_YL44 239840          C      G
## 2 Akkermansia_muciniphila_YL44 241793          A      G
## 3 Akkermansia_muciniphila_YL44 355328          A      T
## 4 Akkermansia_muciniphila_YL44 356291          C      A
## 5 Akkermansia_muciniphila_YL44 2351445         C      T
## 6 Bacteroides_caecimuris_I48 1601848          T      C
##
##              Feature 1683-0
## 1 autotransporter-associated beta strand repeat-containing protein 0.000000
## 2 autotransporter-associated beta strand repeat-containing protein 0.049587
## 3                               protein kinase 0.138182
## 4                               protein kinase 0.000000
## 5 protein phosphatase 2C domain-containing protein 0.000000
## 6                               TonB-dependent receptor 0.041609
## 1683-4 1683-9
## 1 0.000000 0.000000
## 2 0.031414 0.076271
## 3 0.132275 0.138211
## 4 0.000000 0.000000
## 5 0.000000 0.000000
## 6 0.000000 0.038251
```

Explain that every row is a SNP call with many sample columns (e.g., 1683-0).

3. Helper to build wide + long subsets

```
build_subset <- function(genomes, wide_out, long_out) {
  message('Preparing subset for genomes: ', paste(genomes, collapse = ', '))
  subset_df <- raw_df[raw_df$Genome %in% genomes, , drop = FALSE]
  if (nrow(subset_df) == 0) {
    stop('No rows left after filtering for ', paste(genomes, collapse = ', '))
  }

  subset_df$snp_id <- paste(subset_df$Position,
                           subset_df$Alternative,
                           subset_df$Reference,
                           sep = '-')

  sample_cols <- setdiff(names(subset_df),
                        c('Genome', 'snp_id', 'Position',
                          'Alternative', 'Reference', 'Feature'))

  wide_df <- subset_df[, c('Genome', 'snp_id', 'Position', sample_cols), drop = FALSE]
  write.csv(wide_df, wide_out, row.names = FALSE)
  message('Saved: ', wide_out)

  long_parts <- list()
```

```

for (col in sample_cols) {
  long_parts[[length(long_parts) + 1]] <- data.frame(
    Genome = wide_df$Genome,
    snp_id = wide_df$snp_id,
    Position = wide_df$Position,
    sample = col,
    value = wide_df[[col]],
    stringsAsFactors = FALSE
  )
}

long_df <- do.call(rbind, long_parts)
split_ids <- strsplit(long_df$sample, '-', fixed = TRUE)
long_df$mouse_id <- vapply(split_ids, function(x) x[[1]], character(1))
long_df$day <- as.integer(vapply(split_ids,
                                function(x) if (length(x) >= 2) x[[2]] else NA_character_,
                                character(1)))

long_df$sample <- NULL

# Attach treatment group information (if available)
if (file.exists(group_map_path)) {
  group_map <- read.csv(group_map_path, stringsAsFactors = FALSE, check.names = FALSE)
  group_map$Mouse <- as.character(group_map$Mouse)
  group_map$Day <- as.integer(group_map$Day)
  long_df <- merge(
    long_df,
    group_map[, c('Mouse', 'Day', 'Group')],
    by.x = c('mouse_id', 'day'),
    by.y = c('Mouse', 'Day'),
    all.x = TRUE
  )
  names(long_df)[names(long_df) == 'Group'] <- 'treatment_group'
  long_df <- long_df[, c('Genome', 'snp_id', 'Position', 'value', 'mouse_id', 'day', 'treatment_group')]
}

write.csv(long_df, long_out, row.names = FALSE)
message('Saved: ', long_out)

list(wide = wide_df, long = long_df)
}

```

4. Dataset 1 – two spotlight genomes

```

primary_genomes <- c('Akkermansia_muciniphila_YL44', 'Bacteroides_caecimuris_I48')
dataset1 <- build_subset(primary_genomes, subset_path, long_path)

## Preparing subset for genomes: Akkermansia_muciniphila_YL44, Bacteroides_caecimuris_I48
## Saved: dataset1_subset.csv
## Saved: dataset1_subset_long.csv
head(dataset1$wide[, 1:5])

##           Genome      snp_id Position 1683-0 1683-4

```

```
## 1 Akkermansia_muciniphila_YL44 239840-C-G 239840 0.000000 0.000000
## 2 Akkermansia_muciniphila_YL44 241793-A-G 241793 0.049587 0.031414
## 3 Akkermansia_muciniphila_YL44 355328-A-T 355328 0.138182 0.132275
## 4 Akkermansia_muciniphila_YL44 356291-C-A 356291 0.000000 0.000000
## 5 Akkermansia_muciniphila_YL44 2351445-C-T 2351445 0.000000 0.000000
## 6 Bacteroides_caecimuris_I48 1601848-T-C 1601848 0.041609 0.000000
```

```
head(dataset1$long)
```

```
##           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
```

5. Dataset 2 – add Turicimonas for exercises

```
extended_genomes <- c('Akkermansia_muciniphila_YL44',
                      'Bacteroides_caecimuris_I48',
                      'Turicimonas_muris_YL45')
dataset2 <- build_subset(extended_genomes, subset2_path, long2_path)
```

```
## Preparing subset for genomes: Akkermansia_muciniphila_YL44, Bacteroides_caecimuris_I48, Turicimonas_muris_YL45
```

```
## Saved: dataset2_subset.csv
```

```
## Saved: dataset2_subset_long.csv
```

```
head(dataset2$long)
```

```
##           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
```

With both CSV sets created, move on to `scripts/01_explore_data.Rmd` for the guided walkthrough, or try the exercises notebook (`scripts/01_explore_data_exercises.Rmd`, with `scripts/01_explore_data_exercises_solution.Rmd`).

as a key) that uses the three-genome dataset, and finally render `scripts/02_simple_heatmap.Rmd` for the visualization.