EDA_preprocessingR

Louise Buijs

2025-3-11

The purpose of this preprocessing step is to go from raw gene expression counts to a smaller subset of genes that are of particular interest wrt the study treatment (a dietary intervention). The goal of the project is to uplevel the practice of finding rules in gene expression data (‘gene1 increased -> genes 2 and 3 increased’) by incorporating semantic informations such as pathways and intracellular locations into the rule mining process.

The dataset has been used in multiple studies before so all of this EDA stuff is already known

FOR NOW THE ONLY IMPORTANT THING TO DO IS TO ANNOTATE THE RAW DATA WITH GENE NAMES AND SELECT THE INTERESTING ONES BECAUSE RULE MINING ON A BIG NUMBER OF GENES IS COMPUTATIONALLY IMPOSSIBLE AND UNINTERESTING

Read in raw counts and metadata

```
# READ THE FILE
geo_lines <- readLines("GSE77962_series_matrix.txt")

# Extract only the metadata lines (those starting with "!Sample_")
metadata_lines <- geo_lines[grep("^!Sample_", geo_lines)]

# Convert to a dataframe
metadata_df <- read.table(text = metadata_lines, sep = "\t", header = FALSE, stringsAsFactors =
#metadata_df <- metadata_df[1:20,]
headers = metadata_df[,1]
headers
```

```
##  [1] "!Sample_title"                   "!Sample_geo_accession"
##  [3] "!Sample_status"                  "!Sample_submission_date"
##  [5] "!Sample_last_update_date"        "!Sample_type"
##  [7] "!Sample_channel_count"           "!Sample_source_name_ch1"
##  [9] "!Sample_organism_ch1"            "!Sample_characteristics_ch1"
## [11] "!Sample_characteristics_ch1"     "!Sample_characteristics_ch1"
## [13] "!Sample_characteristics_ch1"     "!Sample_characteristics_ch1"
## [15] "!Sample_characteristics_ch1"     "!Sample_characteristics_ch1"
## [17] "!Sample_characteristics_ch1"     "!Sample_characteristics_ch1"
## [19] "!Sample_characteristics_ch1"     "!Sample_treatment_protocol_ch1"
## [21] "!Sample_growth_protocol_ch1"     "!Sample_molecule_ch1"
## [23] "!Sample_extract_protocol_ch1"    "!Sample_label_ch1"
## [25] "!Sample_label_protocol_ch1"      "!Sample_taxid_ch1"
## [27] "!Sample_hyb_protocol"            "!Sample_scan_protocol"
## [29] "!Sample_description"             "!Sample_data_processing"
## [31] "!Sample_platform_id"             "!Sample_contact_name"
## [33] "!Sample_contact_email"           "!Sample_contact_laboratory"
## [35] "!Sample_contact_department"      "!Sample_contact_institute"
## [37] "!Sample_contact_address"         "!Sample_contact_city"
## [39] "!Sample_contact_zip/postal_code" "!Sample_contact_country"
## [41] "!Sample_supplementary_file"      "!Sample_data_row_count"
```

The titles of the columns aren't very informative and most of it is also not interesting for the analysis. If we look at the first entries it is clear what their names should be. Lets look at sample characteristics columns 10-20

```r
# Transpose the data (excluding the header) so that we get samples as rows
metadata_t <- as.data.frame(t(metadata_df[, -1]), stringsAsFactors = FALSE)
colnames(metadata_t) <- headers
rownames(metadata_t) <- metadata_t[,'!Sample_geo_accession']

metadata_t[0:5, 10:20]
```

```
##            !Sample_characteristics_ch1
## GSM2062466               subject_id: 1
## GSM2062467               subject_id: 1
## GSM2062468               subject_id: 1
## GSM2062469               subject_id: 2
## GSM2062470               subject_id: 2
##                           !Sample_characteristics_ch1.1
## GSM2062466 tissue: abdominal subcutaneous white adipose
## GSM2062467 tissue: abdominal subcutaneous white adipose
## GSM2062468 tissue: abdominal subcutaneous white adipose
## GSM2062469 tissue: abdominal subcutaneous white adipose
## GSM2062470 tissue: abdominal subcutaneous white adipose
##               !Sample_characteristics_ch1.2
## GSM2062466 treatment: very-low-calorie diet
## GSM2062467 treatment: very-low-calorie diet
## GSM2062468 treatment: very-low-calorie diet
## GSM2062469 treatment: very-low-calorie diet
## GSM2062470 treatment: very-low-calorie diet
##                     !Sample_characteristics_ch1.3 !Sample_characteristics_ch1.4
## GSM2062466             time point: at study start                   Sex: female
## GSM2062467   time point: after weight loss period                   Sex: female
## GSM2062468 time point: after weight stable period                   Sex: female
## GSM2062469             time point: at study start                   Sex: female
## GSM2062470   time point: after weight loss period                   Sex: female
##            !Sample_characteristics_ch1.5 !Sample_characteristics_ch1.6
## GSM2062466                 age (yrs): 46              height (cm): 167
## GSM2062467                 age (yrs): 46              height (cm): 167
## GSM2062468                 age (yrs): 46              height (cm): 167
## GSM2062469                 age (yrs): 45              height (cm): 164
## GSM2062470                 age (yrs): 45              height (cm): 164
##            !Sample_characteristics_ch1.7 !Sample_characteristics_ch1.8
## GSM2062466            weight (kg): 83.38             bmi (kg/m2): 29.9
## GSM2062467            weight (kg): 75.16            bmi (kg/m2): 26.95
## GSM2062468            weight (kg): 73.79            bmi (kg/m2): 26.46
## GSM2062469            weight (kg): 80.94            bmi (kg/m2): 30.09
## GSM2062470            weight (kg): 71.92            bmi (kg/m2): 26.74
##            !Sample_characteristics_ch1.9
## GSM2062466              body fat %: 39.9
## GSM2062467              body fat %: 35.8
## GSM2062468              body fat %: 32.4
## GSM2062469              body fat %: 46.7
## GSM2062470                body fat %: 43
##
## GSM2062466 The participants in our study followed a dietary intervention program that was di
## GSM2062467 The participants in our study followed a dietary intervention program that was di
## GSM2062468 The participants in our study followed a dietary intervention program that was di
## GSM2062469 The participants in our study followed a dietary intervention program that was di
## GSM2062470 The participants in our study followed a dietary intervention program that was di
```

We can see that: [10] !Sample_characteristics_ch1 = subject_id [11] !Sample_characteristics_ch1.1 = tissue [12] !Sample_characteristics_ch1.2 = treatment [13] !Sample_characteristics_ch1.3 = timepoint [14] !Sample_characteristics_ch1.4 = sex [15] !Sample_characteristics_ch1.5 = age (yrs) [16] !Sample_characteristics_ch1.6 = height [17] !Sample_characteristics_ch1.7 = weight [18] !Sample_characteristics_ch1.8 = bmi [19] !Sample_characteristics_ch1.9 = bodyfat

```
length(unique(metadata_t[,10])) #subject
```

```
## [1] 53
```

```
length(unique(metadata_t[,11])) # tissue
```

```
## [1] 1
```

```
length(unique(metadata_t[,12])) # treatment
```

```
## [1] 2
```

```r
length(unique(metadata_t[,13])) # timepoint
```

```
## [1] 3
```

```r
unique(metadata_t[,12])
```

```
## [1] "treatment: very-low-calorie diet" "treatment: low-calorie diet"
```

```r
unique(metadata_t[,13])
```

```
## [1] "time point: at study start"
## [2] "time point: after weight loss period"
## [3] "time point: after weight stable period"
```

There is 53 subjects, 1 tissue, 2 treatments 3 timepoints.

For simplicity we will focus only of the first two timepoints

it is not the purpose of this thesis to find a great analysis of this particular dataset and take all these factors into account ! Subject_id, treatment and timepoint should be enough to proceed with for now.

```r
## Most of the information is the same for each sample
# Small lookup table with necessary info for now
# Small lookup table with necessary info for now
metadata_small <- metadata_t[,c(10,11,12,13,14)]
colnames(metadata_small) <- c('subject_id', 'tissue', 'treatment','timepoint','sex')
metadata_small$subject_id <- gsub("subject_id: ", "", metadata_small$subject_id)
metadata_small$timepoint <- gsub("time point: ", "", metadata_small$timepoint)
metadata_small$treatment <- gsub("treatment: ", "", metadata_small$treatment)


sum(metadata_small$treatment == 'low-calorie diet')
```

```
## [1] 76
```

```r
# The half of the subjects has gotten very low calory
```

the following block of code is just to annotate the probes (there are multiple probes on a gene expression chip that bind to dna like puzzle pieces, and often multiple probes for one gene so we have to combine these)

```
# Load count matrix (rows = samples, columns = genes)
counts <- read.table("GSE77962_series_matrix.txt", sep = "\t", comment.char = "!", header = TRU


### ANNOTATE PROBES AND AVERAGE BY GENE
# For Affymetrix Human Gene 1.1 ST array
# Install and load the appropriate annotation package
if (!requireNamespace("hugene11sttranscriptcluster.db", quietly = TRUE)) {
  BiocManager::install("hugene11sttranscriptcluster.db")
}
```

```
##
```

```
library(hugene11sttranscriptcluster.db)
```

```
## Loading required package: AnnotationDbi
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, saveRDS, setdiff,
##     table, tapply, union, unique, unsplit, which.max, which.min
```

```
## Loading required package: Biobase
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
## Loading required package: IRanges

## Warning: package 'IRanges' was built under R version 4.4.2

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:utils':
##
##     findMatches

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: org.Hs.eg.db

##
```

```
# Get probe IDs
probe_ids <- rownames(counts)

# Map probes to genes
gene_map <- AnnotationDbi::select(hugene11sttranscriptcluster.db,
                                  keys = probe_ids,
                                  columns = c("SYMBOL", "ENTREZID"),
                                  keytype = "PROBEID")
```

```
## 'select()' returned 1:many mapping between keys and columns
```

```
# Remove NA mappings and duplicates
gene_map <- gene_map[!is.na(gene_map$SYMBOL), ]
gene_map <- gene_map[!duplicated(gene_map$PROBEID), ]

# Check if you have probe IDs or transcript cluster IDs
```

```r
# If your dataset has transcript cluster IDs already:
if (length(intersect(probe_ids, gene_map$PROBEID)) < length(probe_ids) * 0.1) {
  # You might be using transcript cluster IDs instead of probe IDs
  gene_map <- AnnotationDbi::select(hugene11sttranscriptcluster.db,
                                   keys = probe_ids,
                                   columns = c("SYMBOL", "ENTREZID"),
                                   keytype = "TRANSCRIPTCLUSTERID")
  gene_map <- gene_map[!is.na(gene_map$SYMBOL), ]
  gene_map <- gene_map[!duplicated(gene_map$TRANSCRIPTCLUSTERID), ]
}

# Add gene symbols to expression data
counts_with_genes <- merge(gene_map, counts, by.x = "PROBEID", by.y = "row.names")
# If using transcript cluster IDs:
if (ncol(counts_with_genes) < 5) { # Check if merge was successful
  counts_with_genes <- merge(gene_map, counts, by.x = "TRANSCRIPTCLUSTERID", by.y = "row.names"
}

# Average probes by gene (if multiple probes map to the same gene)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:AnnotationDbi':
##
##     select
```

```
## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
```

```
## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union
```

```
## The following object is masked from 'package:Biobase':
##
##     combine
```

```
## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
counts_by_gene <- counts_with_genes %>%
  group_by(SYMBOL) %>%
  summarize(across(!c(PROBEID, ENTREZID), mean, na.rm = TRUE)) %>%
  filter(!is.na(SYMBOL) & SYMBOL != "")
```

```
## Warning: There was 1 warning in `summarize()`.
## i In argument: `across(!c(PROBEID, ENTREZID), mean, na.rm = TRUE)`.
## i In group 1: `SYMBOL = "A1CF"`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```
# If using transcript cluster IDs:
if (!"PROBEID" %in% colnames(counts_with_genes)) {
  counts_by_gene <- counts_with_genes %>%
    group_by(SYMBOL) %>%
    summarize(across(!c(TRANSCRIPTCLUSTERID, ENTREZID), mean, na.rm = TRUE)) %>%
    filter(!is.na(SYMBOL) & SYMBOL != "")
}

# Convert to matrix
gene_expr_matrix <- as.matrix(counts_by_gene[,-1])
rownames(gene_expr_matrix) <- counts_by_gene$SYMBOL
```

```
dim(gene_expr_matrix)
```

```
## [1] 19998   152
```

See here the dimensions of the gene expression matrix

Now that we have the gene expression matrix we need to find a significant subset.

```r
# First, make sure your factors are properly defined
metadata_small$timepoint <- factor(metadata_small$timepoint)
metadata_small$subject_id <- factor(metadata_small$subject_id)


design <- model.matrix(~ 0 + timepoint, data = metadata_small)
colnames(design) <- make.names(colnames(design))

# Fit without accounting for subject correlation
# MAYBE TRY FILTER ONLY ON ONE CONDITION (ADDED LATER)
subjects_treatment1_ids <- metadata_small$treatment == 'very-low-calorie diet'
# Subset the expression matrix to only include those subjects
gene_expr_matrix_filtered <- gene_expr_matrix[, subjects_treatment1_ids]

# Create appropriate design matrix for the filtered data
design <- model.matrix(~ 0 + timepoint, data = metadata_small[subjects_treatment1_ids,])
colnames_design <- make.names(colnames(design))
colnames(design) <-colnames_design


# Fit the model
fit <- lmFit(gene_expr_matrix_filtered, design)
fit <- eBayes(fit)


# Fit without accounting for subject correlation
#fit <- lmFit(gene_expr_matrix, design)
#fit <- eBayes(fit)



# Create and apply contrasts
contrast_matrix <- makeContrasts(
  weight_loss_vs_study_start = timepointafter.weight.loss.period - timepointat.study.start,
  weight_stable_vs_study_start = timepointafter.weight.stable.period - timepointat.study.start
  weight_stable_vs_weight_loss = timepointafter.weight.stable.period - timepointafter.weight.lo
  levels = design
)

fit2 <- contrasts.fit(fit, contrast_matrix)
fit2 <- eBayes(fit2)


# Get results for each comparison
# Weight loss vs study start
results_loss_vs_start <- topTable(fit2, coef=1, number=Inf)
sig_loss_vs_start <- results_loss_vs_start[abs(results_loss_vs_start$logFC) > 1 &
```

```
                                              results_loss_vs_start$adj.P.Val < 0.05, ]

# Weight stable vs study start
results_stable_vs_start <- topTable(fit2, coef=2, number=Inf)
sig_stable_vs_start <- results_stable_vs_start[abs(results_stable_vs_start$logFC) > 1 &
                                                 results_stable_vs_start$adj.P.Val < 0.05, ]

# Weight stable vs weight loss
results_stable_vs_loss <- topTable(fit2, coef=3, number=Inf)
sig_stable_vs_loss <- results_stable_vs_loss[abs(results_stable_vs_loss$logFC) > 1 &
                                               results_stable_vs_loss$adj.P.Val < 0.05, ]

dim(sig_stable_vs_loss)
```

```
## [1] 10  6
```

```
dim(sig_stable_vs_start)
```

```
## [1] 1 6
```

```
dim(sig_loss_vs_start)
```

```
## [1] 20  6
```

```
# Save results to CSV
write.csv(sig_loss_vs_start, "DE_genes_loss_vs_start.csv")
write.csv(sig_stable_vs_start, "DE_genes_stable_vs_start.csv")
write.csv(sig_stable_vs_loss, "DE_genes_stable_vs_loss.csv")
```

What I did is I tried first to find sign genes for the whole dataset, this yielded only 3 genes. Then for the low calory diet, then for the very low calory diet. the last one had the most genes (20).

Unfortunately the results are still too sparse and I think personally that its most interesting to look at different diets so back to whole dataset and lower p value. we are going to mine for rules that include multiple genes anyways and the genes on its own are not too important. mainly the semantic inclusion into rule mining in the python document.

```
### So the results above are too sparse, probably because of individual effects not taken into
### LETS TRY TO INCORPORATE INDIVIDUALS ONE MORE TIME
# Step 1: Make sure factors are set correctly
```

```r
metadata_small$timepoint <- factor(metadata_small$timepoint)
metadata_small$subject_id <- factor(metadata_small$subject_id)

# Step 2: Use duplicate Correlation to account for subject-level correlation
library(statmod)
```

```
## Warning: package 'statmod' was built under R version 4.4.3
```

```r
design <- model.matrix(~0 +timepoint, data = metadata_small)
colnames(design) <- colnames_design

corfit <- duplicateCorrelation(gene_expr_matrix, design, block = metadata_small$subject_id)
consensus_correlation <- corfit$consensus

# Step 3: Fit the model with the correlation structure
fit_cor <- lmFit(gene_expr_matrix, design, block = metadata_small$subject_id,
             correlation = consensus_correlation)

# Step 4: Create specific contrasts for time point comparisons
# If you have time points A, B, C, you might want B vs A, C vs A, and C vs B
# For simplicity, let's assume you want to compare the two time points in your results
contrast_matrix <- makeContrasts(
  comparison = timepointafter.weight.loss.period - timepointat.study.start,
  levels = design
)

# Step 5: Apply the contrast and compute statistics
fit3 <- contrasts.fit(fit_cor, contrast_matrix)
fit3 <- eBayes(fit3)
```

The code below gets me a gene set that are most different between timepoints for most individuals

```r
# Step 6: Get results with the proper fold changes
de_results <- topTable(fit3, number = Inf)
head(de_results)
```

```
##               logFC   AveExpr          t      P.Value    adj.P.Val        B
## FADS2    -1.3361595  6.553256 -11.090664 1.993411e-21 3.986423e-17 37.59288
## SCD      -1.2215850 10.329806  -8.807798 2.355663e-15 2.355427e-11 24.24284
## TNFRSF25 -0.5042736  6.385188  -8.478979 1.646772e-14 1.097738e-10 22.38412
## SRPX      0.5235735  8.741186   8.426115 2.245959e-14 1.122867e-10 22.08751
## FASN     -0.7741949  9.053771  -8.344957 3.611843e-14 1.229266e-10 21.63342
## FADS1    -0.9672848  7.013333  -8.341379 3.688167e-14 1.229266e-10 21.61343
```

```
# Step 7: Filter for significant genes
# I USED A LESS SEVERE FOLD CHANGE HERE BECAUSE I REALLY NEEDED A BIGGER SET OF SIGN GENES
# TO CONTINUE WITH THE RULE MINING ANALYSIS. FOR THE REST OF PROJECT ASK AALTJAN FOR HELP
sig_genes <- de_results[abs(de_results$logFC) > 0.5 & de_results$adj.P.Val < 0.05, ]
nrow(sig_genes)
```

```
## [1] 78
```

```
sig_genes_symbols <- row.names(sig_genes)


# NOW FILTER THE COUNTS WITH GENES MATRIX TO INCLUDE ONLY THOSE GENES AND THOSE TIMEPOINTS
samples_to_keep <- row.names(metadata_small[metadata_small$timepoint != "after weight stable pe
rows_to_keep <- row.names(counts_with_genes[counts_with_genes$SYMBOL %in% sig_genes_symbols, ])

counts_filtered <- counts_with_genes[rows_to_keep, c(colnames(counts_with_genes)[0:3], samples_
write.csv(counts_filtered, 'DE_genes_and_counts_lfc05_start_afwlp.csv')


# convert timepoints to ABC
metadata_small$timepoint_letter <- ifelse(metadata_small$timepoint == "at study start", "A",
                                                                           ifelse(metadata_smal

# Create a unique identifier with underscores inbetween
metadata_small$super_id <- paste(metadata_small$subject_id, metadata_small$timepoint_letter, se

samples_to_keep_super_ids <- metadata_small$super_id[row.names(metadata_small) %in% samples_to_

colnames(counts_filtered) <- c(colnames(counts_with_genes)[0:3], samples_to_keep_super_ids)
write.csv(counts_filtered, 'DE_genes_and_counts_lfc05_start_afwlp_super_ids.csv')

head(counts_filtered)
```

```
##      PROBEID   SYMBOL ENTREZID       1A        1B       2A       2B       3A
## 203  7899160     CD52     1043 8.134457 10.210149 5.861022 7.653125 6.707001
## 464  7902623  DNASE2B    58511 4.158579  7.160408 3.167454 4.475914 3.920600
## 840  7907160   ATP1B1      481 6.759983  8.906521 5.863647 6.595627 6.060681
## 908  7908388     RGS1     5996 4.239624  6.811720 3.174690 4.103133 4.324414
## 1189 7912040 TNFRSF25     8718 7.167506  5.953195 6.760911 6.252471 6.959088
## 1190 7912056 TNFRSF25     8718 6.645332  5.517639 6.289479 5.451551 6.245565
##            3B       4A       4B       5A       5B       6A       6B       8A
## 203  9.037108 6.728515 6.851581 6.336888 8.588973 5.841269 6.214476 6.904097
## 464  5.838410 3.727037 3.424260 3.970920 5.353192 3.505347 4.010981 3.865534
## 840  7.935532 5.569732 6.469043 5.380471 7.739040 5.578104 5.766979 5.219605
## 908  5.359925 3.569452 4.046488 4.464785 6.173643 3.265122 4.127828 3.446786
## 1189 6.270838 6.959360 6.546689 6.041338 5.626827 7.262545 7.284986 7.516614
## 1190 5.056784 6.329256 6.329667 6.051894 4.784949 6.844278 6.386385 6.914055
##            8B       9A       9B      10A      10B      11A      11B      12A
## 203  7.670792 6.710360 7.308627 6.725914 6.480855 8.476643 8.493209 6.072253
## 464  4.956236 3.678474 3.835236 3.891718 3.493240 5.053136 5.419601 4.153485
## 840  6.361325 5.868877 5.935886 5.809937 6.095144 6.787752 6.930536 4.605902
## 908  4.493949 3.947752 4.273026 3.695909 4.564817 5.200577 6.968773 3.675118
## 1189 6.462119 7.073858 6.836313 7.268467 7.207409 6.988007 6.177692 7.501993
## 1190 5.835288 6.233525 6.022816 6.785720 6.539829 6.058373 5.073237 6.672418
##           12B      13A      13B      14A      14B      15B      18A      19A
## 203  6.147011 7.224516 6.313239 6.609828 8.622304 7.123610 6.520185 6.536139
## 464  3.962472 3.974644 3.569311 3.425857 4.310123 3.904699 3.942993 3.322331
## 840  5.809560 5.244641 5.854433 5.730607 6.346555 5.869825 6.080353 5.419933
## 908  3.440262 3.527963 3.771698 3.552204 5.389790 4.314129 4.268627 4.086087
## 1189 7.007691 7.192445 6.788578 7.119969 6.557539 6.564721 6.956910 7.037020
## 1190 6.133263 6.602370 6.233511 6.758310 5.275706 5.714604 6.108697 6.380464
##           19B      20A      20B      21A      21B      22A      22B      23A
## 203  6.909988 7.928412 7.903553 7.775556 5.985852 7.622243 7.126414 7.421563
## 464  3.695671 5.161746 4.679564 3.411213 3.213080 4.895486 4.543006 4.021206
## 840  5.364632 7.107973 6.499171 5.639036 5.450009 6.167284 6.245366 5.922226
## 908  3.661086 5.585445 6.006256 3.987155 4.099541 4.484822 4.935653 4.804419
## 1189 7.497979 6.650508 6.463688 6.278933 6.556213 7.466458 6.987677 7.138571
## 1190 6.662541 6.179687 5.279073 5.997567 6.009214 6.623666 6.109399 6.526600
##           23B      24A      24B      25A      25B      26A      26B      27A
## 203  6.547846 6.627578 6.594856 7.467305 8.745738 6.894725 6.636138 6.632940
## 464  3.876234 3.677432 3.669931 3.540783 4.916396 3.928631 3.582620 3.849147
## 840  5.533976 5.829690 6.161008 5.654271 7.227757 5.543739 5.573092 5.996690
## 908  3.994215 3.904811 3.930051 4.317715 4.799894 3.680915 4.241361 4.011076
## 1189 6.779129 7.181142 6.765441 6.933731 6.543250 6.999697 6.329089 6.932318
## 1190 5.672350 6.113716 5.851751 6.377028 5.292814 6.477786 5.376357 5.748515
##            27B      28A      28B      29A      29B      30A      30B      31A
## 203  10.580799 6.810658 7.230671 7.669014 6.866339 8.729786 8.698156 5.974117
## 464   8.170017 3.695414 4.457136 4.982536 3.744234 5.838808 6.738252 3.545745
## 840   9.211274 5.989795 6.084017 6.351486 5.660254 7.543755 8.361150 5.644284
## 908   7.545450 3.704867 4.532676 5.363389 4.444162 5.850416 7.071538 3.629401
## 1189  5.576399 7.039847 6.426623 6.393664 6.339062 6.725818 6.079580 6.832900
## 1190  4.878577 6.764146 5.278246 5.514752 5.342634 5.767889 5.036754 6.037124
##           31B      33A      33B      34A      34B      35A      35B      36A
## 203  8.606949 8.088002 9.373059 5.845163 7.659963 8.345991 6.457180 6.339573
## 464  5.339741 4.839911 6.406132 3.669509 4.441150 4.445055 3.878838 4.087422
## 840  7.696399 6.748024 8.457762 5.490272 7.041974 6.859175 5.557979 5.985954
## 908  4.644328 5.458801 6.396388 3.442864 4.576490 5.477693 3.858168 4.268892
## 1189 6.588163 6.729225 6.618942 7.028433 6.535089 7.068107 7.286403 7.119071
## 1190 5.542743 5.732711 5.829663 6.775492 5.664168 6.551389 6.626627 6.331189
##           36B      37A      37B      38A      38B      39A      39B      40A
## 203  6.709823 7.323111 8.313683 6.850173 8.940985 5.879836 7.009198 6.269525
## 464  4.132392 4.358790 5.965207 3.728150 5.797227 3.994667 3.983280 3.670018
## 840  6.241073 6.524843 7.576638 6.066814 8.111757 5.489010 6.283480 5.834354
## 908  4.026248 4.396857 5.935756 3.783810 6.292314 3.868013 3.745852 3.545205
## 1189 6.967784 7.278745 6.763769 7.013474 6.060546 7.252372 7.059647 6.854530
## 1190 5.987343 6.375815 6.093146 6.487894 5.116077 6.849915 6.623483 6.104408
##           40B      41A      41B      42A      42B      43A      43B      45A
## 203  7.048245 7.254876 6.762025 6.263817 6.630301 7.124739 7.107182 8.280291
## 464  4.139528 4.471234 3.834066 3.654922 3.328993 3.517861 3.115883 5.639657
## 840  6.263832 6.722686 6.420667 5.339193 5.468633 5.338003 5.437760 7.473922
## 908  4.823285 4.374391 4.509386 3.882610 3.522274 4.039111 3.948185 4.590812
## 1189 6.649589 6.849407 6.240873 6.773704 6.548363 6.594181 6.330959 6.952932
## 1190 5.871935 6.169161 5.292684 5.868453 5.426029 5.585639 5.034937 6.155762
##           45B      46A      46B      47A      47B      48A      48B      49A
## 203  9.616732 6.973557 9.043801 5.750291 7.263554 6.808592 7.502030 7.242718
## 464  6.914565 3.762626 6.097974 3.688800 4.253249 3.681713 4.584277 4.587075
## 840  9.169952 5.858895 8.092434 5.681347 6.795568 5.573328 6.602009 6.707485
## 908  6.551028 4.038805 5.403808 3.977765 4.486327 3.838848 5.835731 4.724185
## 1189 6.544102 7.167940 6.456756 6.889983 6.471177 7.243334 6.828771 7.038200
## 1190 5.637323 6.433022 5.719685 6.367106 5.578743 6.504233 6.199307 6.400817
##           49B      50A      50B      51A      51B      52A      52B      53A
## 203  6.213906 6.612314 6.112042 6.940455 6.553134 6.543599 6.163758 6.666737
## 464  3.716223 4.304179 3.902749 3.717451 3.931278 3.387563 3.786571 3.926459
## 840  5.609883 5.766695 5.942790 5.986754 6.340394 5.369526 5.893642 5.161061
## 908  4.137805 4.510435 3.582557 3.640737 3.812530 3.567330 3.097924 3.691371
## 1189 7.011960 7.562125 7.357835 6.692287 6.238083 7.154964 6.256203 6.895387
## 1190 6.325264 6.836868 6.831774 5.933672 5.293744 6.581393 5.107726 5.826208
##           53B      54B      55A      55B      56A      59A      59B      61A
## 203  7.097803 6.249290 6.600101 7.822554 6.857970 6.646049 6.361763 7.741065
## 464  4.110288 3.499673 3.739524 5.167338 3.887575 3.678486 3.644584 4.925571
## 840  5.991778 5.038457 5.273203 6.839691 6.301970 6.057653 5.832846 6.790010
## 908  3.908669 3.708535 4.280973 5.422587 4.979840 3.694612 3.242359 5.745579
## 1189 6.682914 6.884111 7.149847 6.315190 6.757991 6.619746 6.161798 6.559976
## 1190 5.209278 6.211077 6.753512 5.236845 5.950555 5.931532 5.401665 5.906813
```