T2D: Exercise and Genetic Expression Profile

Program

Francois Collin, Ph.D.

`r format(Sys.time(), '%Y-%m-%d')`

Table of contents

Preamble

Francois Collin
2022-12-12

Development version:

- only the program is displayed within these pages.
- no data is attached to the repository or displayed within the pages.
- no output is displayed within the pages.

Outputs will be included and made available within the program if the associated manuscript is accepted for publication in peer-review journal.

Analysis Environment

The analysis was realized with R (v4.2.0, R Core Team 2022). Note the use of the additional packages downloaded from the RStudio package manager repository (freeze date 2022-05-04, repos) and complemented by Bioconductor packages (release 3.15):

- `BiocParallel`
- `car` for regression model test (version 3.0-13, Fox and Weisberg 2019)
- `cowplot` for figure composition (version 1.1.1, Wilke 2020)
- `DESeq2` for differential expression analysis and variance stabilizing transformation (version 1.36.0, Love, Huber, and Anders 2014)
- `emmeans` for least-Squares Means (LSM) estimations (version 1.7.3, Lenth 2022)
- `flextable` for output table formatting (version 0.7.0, Gohel 2022)
- `ggplot2` for graphics (version 3.3.6, Wickham 2016)
- `hexbin`
- `lintr`
- `missMDA`
- `multcomp`
- `multcompView`

- `MultiAssayExperiment` for management of multi-omics experiment objects (version 1.22.0, Ramos et al. 2017)
- `tidyr` for data wrangling (version 1.2.0, Wickham and Girlich 2022)
- `WGCNA` for gene clustering Langfelder and Horvath (2012).

To install:

- WGCNA
- dynamicTreeCut

The computational environment was containerized into a Docker image build upon `rocker/verse:4.2.0`.

```
  params <- yaml::read_yaml("_prog.yml")
  params$paths |>
    lapply(\(x) if (!dir.exists(x)) dir.create(x))
```

```
$tlg
NULL

$tab
NULL

$grh
NULL

$dta
NULL
```

1 Missingness in adlb and advs

Program 03

Francois Collin
2022-12-12

Data missingness was addressed by a missing-data imputation algorithm, employed to impute missing values while minimizing bias on results (R package `missMDA`, Josse and Husson 2016)); "observed cases" and "imputed data" later refers to the exclusion/inclusion of imputed data.

The analysis was realized with R (v4.2.0, R Core Team 2022). Note the use of the additional package downloaded from the RStudio package manager repository (freeze date 2022-05-04, repos):

- `ggplot2` for graphics (version 3.3.6, Wickham 2016).

1.1 Program settings

```
params <- yaml::read_yaml("_prog.yml")
devtools::load_all("src/pkg/dbs.data")
```

```
i Loading dbs.data
```

```
devtools::load_all("src/pkg/latarnia.utils")
```

```
i Loading latarnia.utils
Loading required package: grid

Loading required package: shiny
```

```
knitr::opts_chunk$set(results = params$knitr$results)
```

1.2 Data Preparation

```
adsl <- dbs.data::adsl
advs <- dbs.data::advs
adlb <- dbs.data::adlb
```

1.3 Figure 03 output 01

```
library(ggplot2)
gg <- rbind(adlb, advs) |>
  ggplot(aes(subjid, paramcd, fill = dtype)) +
  scale_fill_manual(values = c("white", "gray75")) +
  geom_tile(color = "gray50") +
  theme_minimal() +
  theme(
    text = element_text(size = 7),
    axis.title = element_blank(),
    axis.text.x = element_text(angle = 90, hjust = 1),
    legend.position = "none",
    legend.title = element_blank()
  )
```

```
p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 3.1",
      "Colored raster - Data missingness per Subject and Parameter",
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(gg, width = unit(5.2, "inches"), height = unit(3.5, "inches")) |>
  add_footer(
    paste0("Program t2d_03_bds / Env ayup_dbs:", params$dock$version),
    params$version
```

```
  )
```

```
export_as(
  p,
  file = file.path(params$paths$grh, "fig_03_01.pdf"),
  file_graph_alone = file.path(params$paths$grh, "fig_03_01_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_03_01.pdf
```

```
[log] output saved as: ../tlg/graph/fig_03_01_af.pdf (annot. free)
```

```
show_slate(p)
```

1.4 Session Informations

```
sessioninfo::session_info()
```

2 Demographics and Baseline Anthropometrics

Program 08

Francois Collin
2022-12-12

Demographics data characterized the diabetes groups in terms of age, number of training composing the exercise intervention, BMI and diet at baseline. This was completed by a fine description of the baseline anthropometrics leading differences between diabetes groups. Both characterizations relied on one-way analysis of variance, the diabetes effect significance was ruled by a Fisher test, least mean square estimations were obtained for every diabetes group along with their 95% confidence interval, and pairwise difference estimation and significance relied on Tukey's method.

The analysis was realized with R (v4.2.0, R Core Team 2022). Note the use of the additional packages downloaded from the RStudio package manager repository (freeze date 2022-05-04, repos):

- `car` for regression model test (version 3.0-13, Fox and Weisberg 2019)
- `emmeans` for least-Squares Means (LSM) estimations (version 1.7.3, Lenth 2022)
- `flextable` for output table formatting (version 0.7.0, Gohel 2022)
- `tidyr` for data wrangling (version 1.2.0, Wickham and Girlich 2022).

Target:

☒ Table: Demographics and baseline anthropometrics are tested via an Anova.
☒ Supp. Table: Post-hoc estimations / tests by diabetes groups.
☒ Supp. Table: extension of the anova to additional ADVS/ADLB parameters.

Specifications:

- Variable order

```
params <- yaml::read_yaml("_prog.yml")
devtools::load_all("src/pkg/dbs.data")
```

```
i Loading dbs.data
```

```
devtools::load_all("src/pkg/latarnia.utils")
```

```
i Loading latarnia.utils
Loading required package: grid

Loading required package: shiny
```

```
knitr::opts_chunk$set(results = params$knitr$results)
```

```
adsl <- dbs.data::adsl
advs <- dbs.data::advs
adlb <- dbs.data::adlb
```

```
ads <- adlb |>
  rbind(advs) |>
  subset(dtype == "" & avisit == "Baseline") |>
  subset(select = c(subjid, paramcd, avisit, aval, dtype)) |>
  rbind(
    adsl |>
      subset(select = -c(diab, diabcd)) |>
      tidyr::pivot_longer(
        cols = c("age", "trainn"), names_to = "paramcd", values_to = "aval"
      ) |>
      within(dtype <- "") |>
      within(avisit <- "Baseline")
  ) |>
  (\(x) merge(x = adsl[c("subjid", "diabcd")], y = x, by = "subjid"))() |>
  (\(df, fct = "diabcd") {
    df[paste0(fct, "_n")] <- factor_n(df, fct, id = "subjid", sep = " ")
    df
  })() |>
  within(
    paramcd <- factor(
```

```
      paramcd,
      levels = c(
        "age", "trainn", "GLU0", "GLU30", "GLU60", "GLU120",
        "INSULIN0", "INSULIN30", "INSULIN60", "INSULIN120",
        "HBA1C", "HOMAB", "HOMAIR", "MATSUDA",
        "TRIG", "CHOL", "LDL", "HDL", "VO2MAXE", "WEIGHT", "BMI",
        "FATMASS", "BODYFATP", "LEANMASS", "VAT",
        "DCAL", "FATACFR", "DCARBT", "DFATT", "DPROT", "SMMASS",
        "VO2MAXLBM", "VO2MAXML"
      )
    )
  ) |>
  (\(df) df[order(df$paramcd), ])()
```

```
format_pval <- function(x) {
  p <- round(x, 5)
  ifelse(
    test = p < 0.0001,
    yes = "<0.0001",
    no = ifelse(
      test = p < 0.001,
      yes = format(round(p, 4), nsmall = 4),
      no = ifelse(
        test = p < 0.01,
        yes = format(round(p, 3), nsmall = 3),
        no = format(round(p, 2), nsmall = 2)
      )
    )
  )
}

lm_by_paramcd <- function(x,
                          dep_var = "aval",
                          indep_var = "diabcd_n",
                          covariate = NULL) {

  formula <- paste(
    dep_var, "~",
    if (!is.null(covariate)) paste(covariate, "+"),
    indep_var
  )
```

```
  formula <- as.formula(formula)

  lapply(
    x,
    \(x) list(data = x, lm = lm(formula, data = x))
  )
}

make_specs <- function(var) as.formula(paste("~", var))
lsm_by_param <- function(x, indep_var = "diabcd_n") {
  lapply(
    x,
    \(x) {
      mod_em <- emmeans::emmeans(x$lm, specs = make_specs(indep_var))
      y <- as.data.frame(mod_em)
      cbind(
        paramcd = unique(x$data$paramcd),
        y,
        diabcd_f = car::Anova(x$lm)[indep_var, "Pr(>F)"]
      )
    }
  )
}

lsm_pairs_by_param <- function(x, indep_var = "diabcd_n")
  lapply(
    x,
    \(x) {
      mod_em <- emmeans::emmeans(
        x$lm, specs = indep_var, contr = "revpairwise"
      )
      y <- merge(
        as.data.frame(mod_em$contrast)[c("contrast", "p.value")],
        confint(mod_em)$contrasts
      )
      cbind(paramcd = unique(x$data$paramcd), y)
    }
  )
```

2.1 Tab 08 01 - Demographics and Baseline Anthropometrics by Diabetes Group

```
tab_08_01_raw <- ads |>
  subset(
    paramcd %in% c(
      "age", "trainn", "GLU0", "GLU30", "GLU60", "GLU120",
      "INSULIN0", "INSULIN30", "INSULIN60", "INSULIN120",
      "HBA1C", "HOMAB", "HOMAIR", "MATSUDA",
      "TRIG", "CHOL", "LDL", "HDL", "VO2MAXE", "WEIGHT", "BMI",
      "FATMASS", "BODYFATP", "LEANMASS", "VAT"
    )
  ) |>
  (\(x) split(x, f = x$paramcd, drop = TRUE))() |>
  lm_by_paramcd() |>
  lsm_by_param() |>
  (\(x) Reduce(rbind, x))()

library(tidyr)
```

```
Attaching package: 'tidyr'

The following object is masked from 'package:testthat':

    matches
```

```
tab_08_01 <- tab_08_01_raw |>
  (\(df) df[order(df$diabcd_n), ])() |>
  within({
    val <- paste0(
      signif(emmean, 3),
      " (", signif(lower.CL, 3), ", ", signif(upper.CL, 3), ")"
    )
    pval <- format_pval(diabcd_f)
  }) |>
  pivot_wider(
    id_cols = c("paramcd", "pval", "diabcd_f"),
    values_from = "val",
    names_from = "diabcd_n"
```

```
  )
tab_08_01
```

```
library(flextable)
```

```
Attaching package: 'flextable'

The following objects are masked from 'package:latarnia.utils':

    add_footer, add_header
```

```
tab_08_01_ft <- tab_08_01 |>
  subset(select = -diabcd_f) |>
  flextable() |>
  autofit() |>
  add_header_lines(wrap_long_lines(
    "Analysis Set: Full Analysis Set - Observed Cases at baseline"
  )) |>
  set_caption(
    caption = wrap_long_lines(
      "Tab 08 01 - Analysis of Variance / Least Means Square estimations
      (95% Confidence Interval) of Demographics Parameters and Baseline
      Anthropometrics by Diabetes Group"
    )
  ) |>
  footnote(
    part = "header",
    i = 2, j = 2,
    value = as_paragraph(
      "Note: pval, p value of diabetes group effect test by F test."
    ),
    ref_symbols = "a"
  ) |>
  footnote(
    value = as_paragraph(
      "Source: ADSL and ADVS/ADLB observed cases at baseline."
    ),
    ref_symbols = ""
  ) |>
```

```
  theme_booktabs()
tab_08_01_ft
```

```
Warning: Warning: fonts used in `flextable` are ignored because the `pdflatex`
engine is used and not `xelatex` or `lualatex`. You can avoid this warning
by using the `set_flextable_defaults(fonts_ignore=TRUE)` command or use a
compatible engine by defining `latex_engine: xelatex` in the YAML header of the
R Markdown document.
```

```
bnm <- "tab_08_01"
dir_tab <- params$paths$tab
dir_dta <- params$paths$dta

file.path(dir_tab, paste0(bnm, "_ft.RData")) %T>%
  message("[output] Table saved as ", .) %>%
  save(tab_08_01_ft, file = .)
```

```
[output] Table saved as ../tlg/tables/tab_08_01_ft.RData
```

```
file.path(dir_dta, paste0(bnm, ".RData")) %T>%
  message("[output] Table saved as ", .) %>%
  save(tab_08_01, file = .)
```

```
[output] Table saved as ../data/tab_08_01.RData
```

```
file.path(dir_tab, paste(bnm, sep = ".", "docx")) %T>%
  message("[output] Table saved as ", .) %>%
  save_as_docx(tab_08_01_ft, path = .)
```

```
[output] Table saved as ../tlg/tables/tab_08_01.docx
```

```
file.path(dir_tab, paste(bnm, sep = ".", "html")) %T>%
  message("[output] Table saved as ", .) %>%
  save_as_html(tab_08_01_ft, path = .)
```

```
[output] Table saved as ../tlg/tables/tab_08_01.html
```

```r
file.path(dir_dta, paste(bnm, sep = ".", "csv")) %T>%
  message("[output] Table saved as ", .) %>%
  write.csv(tab_08_01, file = ., row.names = FALSE)
```

```
[output] Table saved as ../data/tab_08_01.csv
```

2.2 Tab 08 02 - Post-hoc: Demographics and Baseline Anthropometrics by Diabetes Group

```r
tab_08_02_raw <- ads |>
  subset(
    paramcd %in% c(
      "age", "trainn", "GLU0", "GLU30", "GLU60", "GLU120",
      "INSULIN0", "INSULIN30", "INSULIN60", "INSULIN120",
      "HBA1C", "HOMAB", "HOMAIR", "MATSUDA",
      "TRIG", "CHOL", "LDL", "HDL", "VO2MAXE", "WEIGHT", "BMI",
      "FATMASS", "BODYFATP", "LEANMASS", "VAT"
    )
  ) |>
  (\(x) split(x, f = x$paramcd, drop = TRUE))() |>
  lm_by_paramcd(indep_var = "diabcd") |>
  lsm_pairs_by_param(indep_var = "diabcd")  |>
  (\(x) Reduce(rbind, x))()

library(tidyr)
tab_08_02 <- tab_08_02_raw |>
  subset(select = c(
    paramcd, contrast, estimate, SE, df, p.value, lower.CL, upper.CL
  ))

library(flextable)
tab_08_02_ft <- tab_08_02 |>
  (\(x) split(x, f = x$paramcd))() |>
  lapply(
    \(x) {
      x$p.value <- format(round(x$p.value, 5))
      x$estimate <- format(signif(x$estimate, 5))
      x$SE <- format(signif(x$SE, 6))
      x$lower.CL <- format(signif(x$lower.CL, 5))
```

```
      x$upper.CL <- format(signif(x$upper.CL, 5))
      x
    }) |>
  (\(x) Reduce(rbind, x))()|>
  flextable() |>
  fontsize(size = 9, part = "all") |>
  autofit() |>
  add_header_lines(
    "Analysis Set: Full Analysis Set - Observed Cases at baseline"
  ) |>
  set_caption(
    caption = wrap_long_lines(
      "Tab 08 02 - Post-hoc tests for the Analysis of Variance of
      Demographics and Baseline Anthropometrics by Diabetes Group"
    )
  ) |>
  add_footer_lines(c(
    "CL, 95% Confidence Limit; SE, Standard Error.",
    wrap_long_lines(
      "Note: P value adjustment by Tukey's method for comparing a family of
      3 estimates."
    ),
    "Source: ADSL and ADVS/ADLB observed cases at baseline."
  )) |>
  theme_booktabs()

tab_08_02_ft
```

```
Warning: Warning: fonts used in `flextable` are ignored because the `pdflatex`
engine is used and not `xelatex` or `lualatex`. You can avoid this warning
by using the `set_flextable_defaults(fonts_ignore=TRUE)` command or use a
compatible engine by defining `latex_engine: xelatex` in the YAML header of the
R Markdown document.
```

```
bnm <- "tab_08_02"
dir_tab <- params$paths$tab
dir_dta <- params$paths$dta

file.path(dir_tab, paste0(bnm, "_ft.RData")) %T>%
  message("[output] Table saved as ", .) %>%
  save(tab_08_02_ft, file = .)
```

```
[output] Table saved as ../tlg/tables/tab_08_02_ft.RData
```

```r
file.path(dir_dta, paste0(bnm, ".RData")) %T>%
  message("[output] Table saved as ", .) %>%
  save(tab_08_02, file = .)
```

```
[output] Table saved as ../data/tab_08_02.RData
```

```r
file.path(dir_tab, paste(bnm, sep = ".", "docx")) %T>%
  message("[output] Table saved as ", .) %>%
  save_as_docx(tab_08_02_ft, path = .)
```

```
[output] Table saved as ../tlg/tables/tab_08_02.docx
```

```r
file.path(dir_tab, paste(bnm, sep = ".", "html")) %T>%
  message("[output] Table saved as ", .) %>%
  save_as_html(tab_08_02_ft, path = .)
```

```
[output] Table saved as ../tlg/tables/tab_08_02.html
```

```r
file.path(dir_dta, paste(bnm, sep = ".", "csv")) %T>%
  message("[output] Table saved as ", .) %>%
  write.csv(tab_08_02, file = ., row.names = FALSE)
```

```
[output] Table saved as ../data/tab_08_02.csv
```

2.3 Tab 08 03 - Demographics and Baseline Anthropometrics by Diabetes Group (additional parameters)

```r
tab_08_03_raw <- ads |>
  subset(
    !paramcd %in% c(
      "age", "trainn", "GLU0", "GLU30", "GLU60", "GLU120",
      "INSULIN0", "INSULIN30", "INSULIN60", "INSULIN120",
```

```
      "HBA1C", "HOMAB", "HOMAIR", "MATSUDA",
      "TRIG", "CHOL", "LDL", "HDL", "VO2MAXE", "WEIGHT", "BMI",
      "FATMASS", "BODYFATP", "LEANMASS", "VAT"
    )
  ) |>
  (\(x) split(x, f = x$paramcd, drop = TRUE))() |>
  lm_by_paramcd() |>
  lsm_by_param() |>
  (\(x) Reduce(rbind, x))()

library(tidyr)
tab_08_03 <- tab_08_03_raw |>
  (\(df) df[order(df$diabcd_n), ])() |>
  within({
    val <- paste0(
      signif(emmean, 3),
      " (", signif(lower.CL, 3), ", ", signif(upper.CL, 3), ")"
    )
    pval <- format_pval(diabcd_f)
  }) |>
  pivot_wider(
    id_cols = c("paramcd", "pval", "diabcd_f"),
    values_from = "val",
    names_from = "diabcd_n"
  )
tab_08_03
```

```
library(flextable)
tab_08_03_ft <- tab_08_03 |>
  subset(select = -diabcd_f) |>
  flextable() |>
  autofit() |>
  add_header_lines(wrap_long_lines(
    "Analysis Set: Full Analysis Set - Observed Cases at baseline"
  )) |>
  set_caption(
    caption = wrap_long_lines(
      "Tab 08 03 - Analysis of Variance / Least Means Square estimations
      (95% Confidence Interval) of Demographics Parameters and Baseline
      Anthropometrics by Diabetes Group for Supplementary Parameters"
    )
```

```
  ) |>
  footnote(
    part = "header",
    i = 2, j = 2,
    value = as_paragraph(
      "Note: pval, p value of diabetes group effect test by F test."
    ),
    ref_symbols = "a"
  ) |>
  add_footer_lines("Source: ADSL and ADVS/ADLB observed cases at baseline.") |>
  theme_booktabs()
tab_08_03_ft
```

```
Warning: Warning: fonts used in `flextable` are ignored because the `pdflatex`
engine is used and not `xelatex` or `lualatex`. You can avoid this warning
by using the `set_flextable_defaults(fonts_ignore=TRUE)` command or use a
compatible engine by defining `latex_engine: xelatex` in the YAML header of the
R Markdown document.
```

```
bnm <- "tab_08_03"
dir_tab <- params$paths$tab
dir_dta <- params$paths$dta

file.path(dir_tab, paste0(bnm, "_ft.RData")) %T>%
  message("[output] Table saved as ", .) %>%
  save(tab_08_03_ft, file = .)
```

```
[output] Table saved as ../tlg/tables/tab_08_03_ft.RData
```

```
file.path(dir_dta, paste0(bnm, ".RData")) %T>%
  message("[output] Table saved as ", .) %>%
  save(tab_08_03, file = .)
```

```
[output] Table saved as ../data/tab_08_03.RData
```

```
file.path(dir_tab, paste(bnm, sep = ".", "docx")) %T>%
  message("[output] Table saved as ", .) %>%
  save_as_docx(tab_08_03_ft, path = .)
```

```
[output] Table saved as ../tlg/tables/tab_08_03.docx
```

```r
file.path(dir_tab, paste(bnm, sep = ".", "html")) %T>%
  message("[output] Table saved as ", .) %>%
  save_as_html(tab_08_03_ft, path = .)
```

```
[output] Table saved as ../tlg/tables/tab_08_03.html
```

```r
file.path(dir_dta, paste(bnm, sep = ".", "csv")) %T>%
  message("[output] Table saved as ", .) %>%
  write.csv(tab_08_03, file = ., row.names = FALSE)
```

```
[output] Table saved as ../data/tab_08_03.csv
```

2.4 Tab 08 04 - Post-hoc: Demographics and Baseline Anthropometrics by Diabetes Group (additional parameters)

```r
tab_08_04_raw <- ads |>
  subset(
    ! paramcd %in% c(
      "age", "trainn", "GLU0", "GLU30", "GLU60", "GLU120",
      "INSULIN0", "INSULIN30", "INSULIN60", "INSULIN120",
      "HBA1C", "HOMAB", "HOMAIR", "MATSUDA",
      "TRIG", "CHOL", "LDL", "HDL", "VO2MAXE", "WEIGHT", "BMI",
      "FATMASS", "BODYFATP", "LEANMASS", "VAT"
    )
  ) |>
  (\(x) split(x, f = x$paramcd, drop = TRUE))() |>
  lm_by_paramcd(indep_var = "diabcd") |>
  lsm_pairs_by_param(indep_var = "diabcd") |>
  (\(x) Reduce(rbind, x))()

library(tidyr)
tab_08_04 <- tab_08_04_raw |>
  subset(select = c(
    paramcd, contrast, estimate, SE, df, p.value, lower.CL, upper.CL
  ))
```

```r
library(flextable)
tab_08_04_ft <- tab_08_04 |>
  (\(x) split(x, f = x$paramcd))() |>
  lapply(
    \(x) {
      x$p.value <- format(round(x$p.value, 5))
      x$estimate <- format(signif(x$estimate, 5))
      x$SE <- format(signif(x$SE, 6))
      x$lower.CL <- format(signif(x$lower.CL, 5))
      x$upper.CL <- format(signif(x$upper.CL, 5))
      x
    }) |>
  (\(x) Reduce(rbind, x))()|>
  flextable() |>
  fontsize(size = 9, part = "all") |>
  autofit() |>
  add_header_lines(
    "Analysis Set: Full Analysis Set - Observed Cases at baseline"
  ) |>
  set_caption(
    caption = wrap_long_lines(
      "Tab 08 04 - Post-hoc tests for the Analysis of Variance of
      Demographics and Baseline Anthropometrics by Diabetes Group"
    )
  ) |>
  footnote(
    value = as_paragraph(wrap_long_lines(
      "CL, 95% Confidence Limit; SE, Standard Error."
    )),
    ref_symbols = ""
  ) |>
  footnote(
    value = as_paragraph(wrap_long_lines(
      "Note: P value adjustment by Tukey's method for comparing a family of
      3 estimates."
    )),
    ref_symbols = ""
  )|>
  footnote(
    value = as_paragraph(wrap_long_lines(
      "Source: ADSL and ADVS/ADLB observed cases at
```

```
      baseline."
    )),
    ref_symbols = ""
  ) |>
  theme_booktabs()

tab_08_04_ft
```

```
Warning: Warning: fonts used in `flextable` are ignored because the `pdflatex`
engine is used and not `xelatex` or `lualatex`. You can avoid this warning
by using the `set_flextable_defaults(fonts_ignore=TRUE)` command or use a
compatible engine by defining `latex_engine: xelatex` in the YAML header of the
R Markdown document.
```

```
bnm <- "tab_08_04"
dir_tab <- params$paths$tab
dir_dta <- params$paths$dta

file.path(dir_tab, paste0(bnm, "_ft.RData")) %T>%
  message("[output] Table saved as ", .) %>%
  save(tab_08_04_ft, file = .)
```

```
[output] Table saved as ../tlg/tables/tab_08_04_ft.RData
```

```
file.path(dir_dta, paste0(bnm, ".RData")) %T>%
  message("[output] Table saved as ", .) %>%
  save(tab_08_04, file = .)
```

```
[output] Table saved as ../data/tab_08_04.RData
```

```
file.path(dir_tab, paste(bnm, sep = ".", "docx")) %T>%
  message("[output] Table saved as ", .) %>%
  save_as_docx(tab_08_04_ft, path = .)
```

```
[output] Table saved as ../tlg/tables/tab_08_04.docx
```

```
  file.path(dir_tab, paste(bnm, sep = ".", "html")) %T>%
    message("[output] Table saved as ", .) %>%
    save_as_html(tab_08_04_ft, path = .)
```

```
[output] Table saved as ../tlg/tables/tab_08_04.html
```

```
  file.path(dir_dta, paste(bnm, sep = ".", "csv")) %T>%
    message("[output] Table saved as ", .) %>%
    write.csv(tab_08_04, file = ., row.names = FALSE)
```

```
[output] Table saved as ../data/tab_08_04.csv
```

2.5 Session Informations

```
  sessioninfo::session_info()
```

3 Anthropometrics Changes From Baseline

Program 09

Francois Collin
2022-12-12

Diabetes group estimations in anthropometrics changes from baseline were obtained and tested by analysis of covariance models (Ancova), relying once more on the Fisher test; to increase accuracy and statistical power, estimations were adjusted for baseline values (Vickers 2001; Van Breukelen 2006; Committee for Medicinal Products for Human Use (CHMP) 2015; O'Connell et al. 2017).

The analysis was realized with R (v4.2.0, R Core Team 2022). Note the use of the additional packages downloaded from the RStudio package manager repository (freeze date 2022-05-04, repos):

- `car` for regression model test (version 3.0-13, Fox and Weisberg 2019)
- `emmeans` for least-Squares Means (LSM) estimations (version 1.7.3, Lenth 2022)
- `flextable` for output table formatting (version 0.7.0, Gohel 2022)
- `tidyr` for data wrangling (version 1.2.0, Wickham and Girlich 2022).

3.1 Settings

```
cfg_prog <- yaml::read_yaml("_prog.yml")
devtools::load_all("src/pkg/dbs.data")
```

```
i Loading dbs.data
```

```
devtools::load_all("src/pkg/latarnia.utils")
```

```
i Loading latarnia.utils
Loading required package: grid

Loading required package: shiny
```

```
source("R/inches.R")

knitr::opts_chunk$set(results = cfg_prog$knitr$results)
```

3.2 Data

```
adsl <- dbs.data::adsl
advs <- dbs.data::advs
adlb <- dbs.data::adlb
```

```
ads <- adlb |>
  rbind(advs) |>
  subset(basetype == "" & avisit != "Baseline") |>
  subset(select = c(subjid, paramcd, avisit, base, chg)) |>
  (\(x) merge(x = adsl[c("subjid", "diabcd")], y = x, by = "subjid"))() |>
  (\(df, fct = "diabcd") {
    df[paste0(fct, "_n")] <- factor_n(df, fct, id = "subjid", sep = " ")
    df
  })()

head(ads)
```

3.3 Helper functions

```
format_pval <- function(x) {
  p <- round(x, 5)
  ifelse(
    test = p < 0.0001,
    yes = "<0.0001",
    no = ifelse(
      test = p < 0.001,
      yes = format(round(p, 4), nsmall = 4),
```

```
      no = ifelse(
        test = p < 0.01,
        yes = format(round(p, 3), nsmall = 3),
        no = format(round(p, 2), nsmall = 2)
      )
    )
  )
}

lm_by_paramcd <- function(x,
                          dep_var = "aval",
                          indep_var = "diabcd_n",
                          covariate = NULL) {

  formula <- paste(
    dep_var, "~",
    if (!is.null(covariate)) paste(covariate, "+"),
    indep_var
  )
  formula <- as.formula(formula)

  lapply(x, \(x) list(data = x, lm = lm(formula, data = x)))
}

make_specs <- function(var) as.formula(paste("~", var))
lsm_by_param <- function(x, indep_var = "diabcd_n") {
  lapply(
    x,
    \(x) {
      mod_em <- emmeans::emmeans(x$lm, specs = make_specs(indep_var))
      y <- as.data.frame(mod_em)
      cbind(
        paramcd = unique(x$data$paramcd),
        y,
        diabcd_f = car::Anova(x$lm)[indep_var, "Pr(>F)"]
      )
    }
  )
}
```

3.4 Tab 09 01 - Ancova - Anthropometrics Changes from Baseline by Diabetes Group

```
tab_09_01_raw <- ads |>
  (\(x) split(x, f = x$paramcd))() |>
  lm_by_paramcd(dep_var = "chg", covariate = "base", indep_var = "diabcd_n") |>
  lsm_by_param(indep_var = "diabcd_n") |>
  (\(x) Reduce(rbind, x))()

tab_09_01 <- tab_09_01_raw |>
  (\(df) df[order(df$diabcd_n), ])() |>
  within({
    val <- paste0(
      signif(emmean, 3),
      " (", signif(lower.CL, 3), ", ", signif(upper.CL, 3), ")"
    )
    pval <- format_pval(diabcd_f)
  }) |>
  tidyr::pivot_wider(
    id_cols = c("paramcd", "pval", "diabcd_f"),
    values_from = "val",
    names_from = "diabcd_n"
  ) |>
  (\(x) x[order(x$diabcd_f), ])()
tab_09_01
```

```
library(flextable)
```

```
Attaching package: 'flextable'
```

```
The following objects are masked from 'package:latarnia.utils':

    add_footer, add_header
```

```
wrap_line <- function(x) paste(strwrap(x, width = 80), collapse = " ")
tab_09_01_ft <- tab_09_01 |>
  subset(select = -diabcd_f) |>
  flextable() |>
```

```
  autofit() |>
  footnote(
    part = "header",
    i = 1, j = 2,
    value = as_paragraph(
      "Note: pval, p value of diabetes group effect test by F test."
    ),
    ref_symbols = "a"
  ) |>
  add_footer_lines(c(
    wrap_line("Source: Full Analysis Set, observed cases at baseline and post
    intervention."),
    "Note: rows are ordered by increasing p values, most significant on top."
  )) |>
  add_header_lines("Analysis Set: Full Analysis Set - Observed Cases") |>
  set_caption(
    caption =  wrap_long_lines(
      "Tab 09 01 - Analysis of Covariance / Least Means Square estimations of
      Anthropometrics Changes from Baseline by Diagnosis Group at
      Month 3 (95% Confidence Interval) Adjusted for Baseline"
    )
  ) |>
  theme_booktabs()
tab_09_01_ft
```

```
Warning: Warning: fonts used in `flextable` are ignored because the `pdflatex`
engine is used and not `xelatex` or `lualatex`. You can avoid this warning
by using the `set_flextable_defaults(fonts_ignore=TRUE)` command or use a
compatible engine by defining `latex_engine: xelatex` in the YAML header of the
R Markdown document.
```

```
bnm <- "tab_09_01"
dir_tab <- cfg_prog$paths$tab
dir_dta <- cfg_prog$paths$dta

file.path(dir_tab, paste0(bnm, "_ft.RData")) %T>%
  message("[output] Table saved as ", .) %>%
  save(tab_09_01_ft, file = .)
```

```
[output] Table saved as ../tlg/tables/tab_09_01_ft.RData
```

```
  file.path(dir_dta, paste0(bnm, ".RData")) %T>%
    message("[output] Table saved as ", .) %>%
    save(tab_09_01, file = .)
```

```
[output] Table saved as ../data/tab_09_01.RData
```

```
  file.path(dir_tab, paste(bnm, sep = ".", "docx")) %T>%
    message("[output] Table saved as ", .) %>%
    save_as_docx(tab_09_01_ft, path = .)
```

```
[output] Table saved as ../tlg/tables/tab_09_01.docx
```

```
  file.path(dir_tab, paste(bnm, sep = ".", "html")) %T>%
    message("[output] Table saved as ", .) %>%
    save_as_html(tab_09_01_ft, path = .)
```

```
[output] Table saved as ../tlg/tables/tab_09_01.html
```

```
  file.path(dir_dta, paste(bnm, sep = ".", "csv")) %T>%
    message("[output] Table saved as ", .) %>%
    write.csv(tab_09_01, file = ., row.names = FALSE)
```

```
[output] Table saved as ../data/tab_09_01.csv
```

```
  detach(package:flextable)
```

3.5 Example: Glucose 120 - Fig 09 01

```
  library(ggplot2)
  dta <- dbs.data::adlb |>
    subset(
      paramcd == "GLU120" &
        dtype == "" &
        avisit == "Month 3" &
```

```
      basetype == ""
  ) |>
  merge(x = adsl[c("subjid", "diabcd")], y = _, by = "subjid")

dta$lm_pred <- predict(lm(aval ~ base, data = dta))
lim <- range(c(dta$aval, dta$base))

gg1 <- ggplot(dta, aes(base, aval, color = diabcd)) +
  geom_point() +
  ylab("Post exercise intervention") +
  xlab("Baseline") +
  geom_segment(aes(xend = base, yend = lm_pred)) +
  geom_smooth(method = "lm", se = FALSE, color = "black", linetype = 1) +
  scale_color_viridis_d(begin = .1, end = .9, direction = -1) +
  coord_cartesian(xlim = lim, ylim = lim) +
  theme_minimal() +
  theme(legend.position = "top")

gg2 <- ggplot(dta, aes(base, fill = diabcd, color = diabcd)) +
  geom_boxplot(alpha = .5, show.legend = FALSE) +
  coord_cartesian(xlim = lim) +
  ylab("diabcd") +
  scale_color_viridis_d(begin = .1, end = .9, direction = -1) +
  scale_fill_viridis_d(begin = .1, end = .9, direction = -1) +
  theme_minimal() +
  theme(
    legend.position = "top",
    axis.title.x = element_blank(),
    axis.text.y = element_blank(),
    panel.grid = element_blank()
  )

gg <- cowplot::plot_grid(
  gg1, gg2,
  nrow = 2,
  align = "v",
  rel_heights = c(1, .15)
)
```

```
`geom_smooth()` using formula 'y ~ x'
```

```r
p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 9.1",
      strwrap(
        "Scatter plot - Glucose 120 at Month 3 in relation to Baseline by
        treatment group", width = 80
      ),
      "Analysis Set: Full Analysis Set - Observed Cases"
    )
  ) |>
  add_figure(gg, height = inches(5.5), width = inches(4.5))|>
  add_note(c(
    "Note: the table prog 08 output 01 focuses on the x-axis and check for
     differences at baseline, the boxplots gives an overview of these
     baseline variation.",
    "Note: the table tab prog 09 output 01 shows the effect of exercise in
     each group. The p.values is provided and accounts for the diagnostic group
     effect, while taking into account the fact that the higher the value at
     baseline, the higher after the exercise intervention. Actually,
     the analysis of covariance model focus on how much above or below the
     general regression (black line) is positionned a diagnosis group.
     E.g., +20 for T2D indicates that the result after exercise is 20 units
     higher than expected if there were no diagnostic group effect; ... however,
     the confidence interval extend from -13.5 and +53.8 which means that
     i) if we repeated the study, 100 times, it would result 95 times from
     -13.5 to 53.8
     ii) the interval includes 0 so we can't conclude to a significant
     increase or decrease in Glucose 120 due to the exercise intervention."
  )) |>
  add_footer(
    c(
      "Program t2d_09_chg / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )

export_as(
  p,
```

```
  file = file.path(cfg_prog$paths$grh, "fig_09_01.pdf"),
  file_graph_alone = file.path(cfg_prog$paths$grh, "fig_09_01_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_09_01.pdf
```

```
[log] output saved as: ../tlg/graph/fig_09_01_af.pdf (annot. free)
```

```
show_slate(p)
```

3.6 Session Informations

```
sessioninfo::session_info()
```

3.7 Untidied information about analysis of change from baseline

Questions: for the analysis of a post-treatment values, should we analyse the change from baseline or percentage change from baseline? Should we adjust for the baseline?

Back to 2003, in the context of randomized clinical trial, the European Medicines Agency (COMMITTEE FOR PROPRIETARY MEDICINAL PRODUCTS 2003), when the endpoint is studied as a change from baseline, the adjustment for baseline improves the accuracy in comparison to non-baseline adjustment; estimates becomes also equivalent to the standard linear model, the choice of change from baseline analysis or raw value is then only a question of interpretability. They renewed the recommendation in 2015 (Committee for Medicinal Products for Human Use (CHMP) 2015).

From the academic side, the topic was repeatedly studied:

- Van Breukelen (2006): > *"In randomized studies both methods [Anova, no BL adjustment vs Ancova] > are unbiased, but ANCOVA has more power"*
- Liu et al. (2009) also highlighted the benefits of adjustment for baseline as a covariate.
- this was later confirmed by Zhang et al. (2014).
- More recently O'Connell et al. (2017) also defended the superiority of the Ancova-change: $y_i = \beta_0 + \beta_1 X_i + \beta_2 Y_{0i,=BL} + \varepsilon_i$

> _"Consistent with existing literature, our results demonstrate that each method leads to unbiased treatment effect estimates, and based on precision of estimates, 95% coverage probability, and power, ANCOVA modeling of either change scores or post-treatment score as the outcome, prove to be the most effective._"

Most of the authors above are specifically working on randomized trial, Vickers (2001) also brought some light on the topic, and highlighted in addition that: working with percentage change is generally a bad idea. The extended to a theoretical works also indicated that the percentage change from baseline *"will also fail to protect from bias in the case of baseline imbalance and will lead to an excess of trials with non-normally distributed outcome data"*.

4 RNASeq - Refresher

Program 06

Francois Collin
2022-12-12

```
params <- yaml::read_yaml("_prog.yml")
```

Target:

- ☒ refresh the differential expression analysis technics with DESeq2.
- ☒ upgrade environment for differential expression analysis.
- ☒ evaluate the impact of confounder adjustment on a specific use case: compare miRNA expression between T2D and NGT at baseline.

```
devtools::load_all("src/pkg/dbs.data")
```

```
i Loading dbs.data
```

```
devtools::load_all("src/pkg/latarnia.utils")
```

```
i Loading latarnia.utils
Loading required package: grid

Loading required package: shiny
```

```
knitr::opts_chunk$set(results = params$knitr$results)
library(assertthat)
source("R/ngs.R")
```

4.0.1 Data preparation

```
adsl <- dbs.data::adsl
advs <- dbs.data::advs
adlb <- dbs.data::adlb

#' Subjid and Visit to Sample
#'
subjvis_to_spl <- function(df) paste0(df$subjid, "v", df$avisitn)

ads <- adlb |>
  subset(
    paramcd %in% c(
      "CHOL", "HBA1C", "HDL", "HOMAB", "HOMAIR", "LDL", "MATSUDA", "TRIG"
    )
  ) %>%
  rbind(advs) |>
  subset(select = -c(ct, dtype, param, base, basetype, chg, pchg)) |>
  tidyr::pivot_wider(names_from = "paramcd", values_from = "aval") |>
  (\(df) merge(adsl, df, by = "subjid"))() |>
  (\(df) S4Vectors::DataFrame(df, row.names = subjvis_to_spl(df)))() |>
  (\(df) {
    assertthat::assert_that(all(table(subjvis_to_spl(df)) == 1))
    df
  })()

ads
```

```
rna <- list(# There will be mi-RNA data.
  mrna = dbs.data::mrna_raw,
  premirna = dbs.data::premirna_raw,
  mirna = dbs.data::mirna_raw
)

rna[c("premirna", "mirna")] <- lapply(
  X = rna[c("premirna", "mirna")],
  FUN = format_mirna
)

# Rows represent genes.
rna <- lapply(X = rna, FUN = function(x) y <- x[rowSums(x) > 0, ])
```

```
rna <- lapply(X = rna, as.matrix)
assertthat::assert_that(all(colnames(rna$premirna) == colnames(rna$mirna)))
rna$allmirna <- rbind(rna$premirna, rna$mirna)
library(testthat)
test_that("rna features discriminated in noexpr, expr", {
  lapply(
    X = rna,
    FUN = function(x) expect_true(all(rowSums(x) > 0))
  )
})
```

```
library(MultiAssayExperiment)
```

```
Loading required package: SummarizedExperiment
```

```
Loading required package: MatrixGenerics
```

```
Loading required package: matrixStats
```

```

Attaching package: 'MatrixGenerics'
```

```
The following objects are masked from 'package:matrixStats':

    colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
    colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
    colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
    colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
    colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
    colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
    colWeightedMeans, colWeightedMedians, colWeightedSds,
    colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
    rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
    rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
    rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
    rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
    rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
    rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
    rowWeightedSds, rowWeightedVars
```

```
Loading required package: GenomicRanges

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, 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, setdiff, sort, table, tapply,
    union, unique, unsplit, which.max, which.min

Loading required package: S4Vectors


Attaching package: 'S4Vectors'

The following objects are masked from 'package:base':

    expand.grid, I, unname

Loading required package: IRanges

Loading required package: GenomeInfoDb

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")'.


Attaching package: 'Biobase'

The following object is masked from 'package:MatrixGenerics':

    rowMedians

The following objects are masked from 'package:matrixStats':

    anyMissing, rowMedians
```

```
#' (Sample-)Map Arrays
#'
#' Use the colnames of `x` to deduce the `primary` and `colnames`.
#' This is used to generate the sample mapping between colData and Experiments.
#'
#' @param x (`dataframe`).
#'
#' @note In our case, primary and colnames are equivalent, colnames could
#' be different from primary names when a biological sample has different
#' names in the biological assays (e.g. machine constraint, technical
#' repetitions).
#'
#' @seealso [MultiAssayExperiment::listToMap()]
#' @examples
#' \dontrun{
#' lapply(rna, map_arrays)
#' MultiAssayExperiment::listToMap(lapply(rna, map_arrays))
#' }
#'
map_arrays <- function(x) {
  y <- data.frame(colname = colnames(x))
  y$primary <- y$colname
  y
}
```

```
besd_mae <- MultiAssayExperiment(
  experiments = ExperimentList(rna),
  colData = ads,
  sampleMap = listToMap(lapply(rna, map_arrays))
)

besd_mae
```

4.1 DE: Baseline, all micro RNA, no confounding factor (`dds_1`)

```
ctrl <- yaml::read_yaml("_prog.yml")$rna

ngs_assay <- "allmirna"

filter_for_depth <- function(mae, assay, depth_threshold) {
  mae[, colSums(mae[[ngs_assay]]) > depth_threshold, ]
}

filter_for_visit <- function(mae, visit) {
  mae[, colData(mae)$avisit == visit, ]
}

filter_for_low_expr <- function(mae, assay, cpm_threshold, frac_cols = 1 / 2) {
  # Genes expressed at least cpm_threshold in frac_cols columns
  mae[
    rowSums(cpm(mae[[assay]]) > cpm_threshold) >
      ncol(mae[[assay]]) * frac_cols,
    ,
  ]
}

ads <- besd_mae |>
  (\(mae) mae[ , , ngs_assay])() |>
  filter_for_depth("allmirna", ctrl$depth_threshold[[ngs_assay]]) |>
  filter_for_visit("Baseline") |>
  filter_for_low_expr("allmirna", ctrl$cpm_threshold[[ngs_assay]])
```

```
Warning: 'experiments' dropped; see 'metadata'
```

```
harmonizing input:
  removing 282 sampleMap rows not in names(experiments)
```

```r
ads

dds_1 <- DESeq2::DESeqDataSetFromMatrix(
  countData = ads[[ngs_assay]],
  colData = colData(ads),
  design = stats::formula(~ diabcd)
)

dds_1_res <- DESeq2::DESeq(
  object = dds_1,
  quiet = FALSE, # default: FALSE
  minReplicatesForReplace = 7, # default: 7
  useT = FALSE, # default: FALSE
  minmu = 0.5, # default: 0.5
  parallel = TRUE,
  BPPARAM = BiocParallel::bpparam()
)
```

```
estimating size factors

estimating dispersions

gene-wise dispersion estimates: 2 workers

mean-dispersion relationship

-- note: fitType='parametric', but the dispersion trend was not well captured by the
   function: y = a/x + b, and a local regression fit was automatically substituted.
   specify fitType='local' or 'mean' to avoid this message next time.

final dispersion estimates, fitting model and testing: 2 workers

-- replacing outliers and refitting for 13 genes
-- DESeq argument 'minReplicatesForReplace' = 7
-- original counts are preserved in counts(dds)
```

```
estimating dispersions
```

```
fitting model and testing
```

```
dds_1_de <- DESeq2::results(
  dds_1_res,
  contrast = c("diabcd", test = "T2D", ref = "NGT"),
  pAdjustMethod = ctrl$adj_meth
) |>
  (\(df) {
    df$feature <- rownames(df)
    df
  }) () |>
  within(log_padj <- -1 * log10(padj))

library(ggplot2)
dds_1_gg <-
  dds_1_de |> as.data.frame() |>
  ggplot(mapping = aes(log2FoldChange, log_padj, fill = log10(..count..))) +
  geom_hline(yintercept = -1 * log10(c(0.05, 0.001)), lty = 2, lwd = .5) +
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  annotate(
    geom  = "label", x = -Inf, y = -1 * log10(0.05),
    label = "p = 0.05",
    fill = "white", hjust = "left", size = 2, alpha = 1
  ) +
  annotate(
    geom  = "label", x = -Inf, y = -1 * log10(0.001),
    label = "p = 0.001",
    fill = "white", hjust = "left", size = 2, alpha = 1
  ) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
  stat_bin_hex() +
  scale_fill_gradient(low = "black", high = "gray90") +
  theme_minimal() +
  theme(legend.position = "bottom", asp = 2 / 3)
```

```
dds_1_gg
```

4.2 DE: Baseline, all micro RNA, accounting for Age, BMI, DCAL, Trainn (`dds_2`)

```
ctrl <- yaml::read_yaml("_prog.yml")$rna

ngs_assay <- "allmirna"

filter_for_depth <- function(mae, assay, depth_threshold) {
  mae[, colSums(mae[[ngs_assay]]) > depth_threshold, ]
}

filter_for_visit <- function(mae, visit) {
  mae[, colData(mae)$avisit == visit, ]
}

filter_for_low_expr <- function(mae, assay, cpm_threshold, frac_cols = 1 / 2) {
  # Genes expressed at least cpm_threshold in frac_cols columns
  mae[
    rowSums(cpm(mae[[assay]]) > cpm_threshold) >
      ncol(mae[[assay]]) * frac_cols,
    ,
  ]
}

scale_confounder <- function(mae, confounder) {
  for (i in seq_along(confounder)) {
    cfd <- confounder[i]
    colData(mae)[cfd] <- scale(colData(mae)[cfd])
  }
  mae
}
ads <- besd_mae |>
  (\(mae) mae[ , , ngs_assay])() |>
  filter_for_depth("allmirna", ctrl$depth_threshold[[ngs_assay]]) |>
  filter_for_visit("Baseline") |>
  filter_for_low_expr("allmirna", ctrl$cpm_threshold[[ngs_assay]]) |>
  scale_confounder(confounder = c("age", "trainn", "BMI", "DCAL"))
```

```
Warning: 'experiments' dropped; see 'metadata'

harmonizing input:
```

```
  removing 282 sampleMap rows not in names(experiments)
```

```r
ads

dds_2 <- DESeq2::DESeqDataSetFromMatrix(
  countData = ads[[ngs_assay]],
  colData = colData(ads),
  design = stats::formula(~ age + BMI + trainn + DCAL + diabcd)
)

dds_2_res <- DESeq2::DESeq(
  object = dds_2,
  quiet = FALSE, # default: FALSE
  minReplicatesForReplace = 7, # default: 7
  useT = FALSE, # default: FALSE
  minmu = 0.5, # default: 0.5
  parallel = TRUE,
  BPPARAM = BiocParallel::bpparam()
)
```

```
estimating size factors

estimating dispersions

gene-wise dispersion estimates: 2 workers

mean-dispersion relationship

-- note: fitType='parametric', but the dispersion trend was not well captured by the
   function: y = a/x + b, and a local regression fit was automatically substituted.
   specify fitType='local' or 'mean' to avoid this message next time.

final dispersion estimates, fitting model and testing: 2 workers
```

```r
dds_2_de <- DESeq2::results(
  dds_2_res,
  contrast = c("diabcd", test = "T2D", ref = "NGT"),
  pAdjustMethod = ctrl$adj_meth
) |>
```

```
  (\(df) {
    df$feature <- rownames(df)
    df
  }) () |>
  within(log_padj <- -1 * log10(padj))

library(ggplot2)
dds_2_gg <-
  dds_2_de |> as.data.frame() |>
  ggplot(mapping = aes(log2FoldChange, log_padj, fill = log10(..count..))) +
  geom_hline(yintercept = -1 * log10(c(0.05, 0.001)), lty = 2, lwd = .5) +
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  annotate(
    geom  = "label", x = -Inf, y = -1 * log10(0.05),
    label = "p = 0.05",
    fill = "white", hjust = "left", size = 2, alpha = 1
  ) +
  annotate(
    geom  = "label", x = -Inf, y = -1 * log10(0.001),
    label = "p = 0.001",
    fill = "white", hjust = "left", size = 2, alpha = 1
  ) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
  stat_bin_hex() +
  scale_fill_gradient(low = "black", high = "gray90") +
  theme_minimal() +
  theme(legend.position = "bottom", asp = 2 / 3)
```

```
dds_2_gg
```

4.3 Comparison with/without confounding factors

```
theme_fun <- function(...) {
  theme_minimal() +
    theme(
      title = element_text(size = 9),
      text = element_text(size = 9)
    ) +
```

```
    theme(...)
}

gg_1_2 <- rbind(
  within(as.data.frame(dds_1_de), facet <- "No confounding factors"),
  within(as.data.frame(dds_2_de), facet <- "~Age + BMI + DCAL + Train")
) |>
  ggplot(mapping = aes(log2FoldChange, log_padj, fill = log10(..count..))) +
  geom_hline(yintercept = -1 * log10(c(0.05, 0.001)), lty = 2, lwd = .5) +
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  annotate(
    geom  = "label", x = -Inf, y = -1 * log10(0.05),
    label = "p = 0.05",
    fill = "white", hjust = "left", size = 2, alpha = 1
  ) +
  annotate(
    geom  = "label", x = -Inf, y = -1 * log10(0.001),
    label = "p = 0.001",
    fill = "white", hjust = "left", size = 2, alpha = 1
  ) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
  stat_bin_hex() +
  scale_fill_gradient(low = "black", high = "gray90") +
  facet_wrap(facet ~ ., ncol = 2) +
  theme_fun(
    legend.position = "bottom"
  ) +
  theme(
    legend.key.width = unit(5, "lines"),
    legend.key.height = unit(.8, "lines")
  )

res <- merge(
  as.data.frame(dds_1_de),
  as.data.frame(dds_2_de),
  by = "feature",
  all = TRUE,
  suffixes = c(".asis", ".cfd")
)
```

```r
fun_label <- function(df,
                      x = "log2FoldChange.asis",
                      y = "log2FoldChange.cfd") {
  cor_fun <- function(meth = "pearson") {
    round(cor(df[[x]], df[[y]], method = meth), 2)
  }
  paste0(
    "atop(",
    "r == ", cor_fun(), ",",
    "rho == ", cor_fun("spearman"),
    ")"
  )
}

lim <- range(unlist(res[c("log2FoldChange.asis", "log2FoldChange.cfd")]))
gg_cor_lfc <- ggplot(res, aes(log2FoldChange.asis, log2FoldChange.cfd)) +
  geom_hex() +
  scale_fill_viridis_c(option = "F", begin = .1, end = .9) +
  geom_abline(slope = 1, intercept = 0, col = "red") +
  annotate(
    "label", x = -Inf, y = Inf, hjust = 0, vjust = 1,
    label = fun_label(res),
    parse = TRUE,
    family = "mono",
    size = 3
  )+
  coord_cartesian(xlim = lim, ylim = lim) +
  labs(
    title = "Log Fold Change (LFC)",
    subtitle = "With / Without Adjustment for Confounding Factors",
    x = "No Adjustment",
    y = "Adjusted for Confounding Factors"
  ) +
  theme_fun(asp = 1)

lim <- range(unlist(res[c("log_padj.asis", "log_padj.cfd")]))
gg_cor_pval <- ggplot(res, aes(log_padj.asis, log_padj.cfd)) +
  geom_hex() +
  scale_fill_viridis_c(option = "D", begin = .1, end = .9) +
  geom_abline(slope = 1, intercept = 0, col = "green2") +
  annotate(
```

```
    "label", x = -Inf, y = Inf, hjust = 0, vjust = 1,
    label = fun_label(res, "log_padj.asis", "log_padj.cfd"),
    parse = TRUE,
    family = "mono",
    size = 3
  )+
  coord_cartesian(xlim = lim, ylim = lim, clip = "off") +
  labs(
    title = expression("Significance: "*-1 %.% log10(padj)),
    subtitle = "With / Without Adjustment for Confounding Factors",
    x = "No Adjustment",
    y = "Adjusted for Confounding Factors"
  ) +
  theme_fun(asp = 1)

library(cowplot)
p <- plot_grid(
  plot_grid(gg_1_2) + theme(plot.background = element_rect(color = "black")),
  plot_grid(
    plot_grid(gg_cor_lfc) +
      theme(plot.background = element_rect(color = "black")),
    plot_grid(gg_cor_pval) +
      theme(plot.background = element_rect(color = "black")),
    labels = c("B", "C")
  ),
  ncol = 1, rel_heights = c(3, 2),
  labels = c("A", NA)
)


p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 6.1",
      strwrap(
        "Volcano plot - Level and significance of Differential Expression among
        all miRNA at Baseline between T2D and NGT", width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
```

```
  ) |>
  add_note(c(
    "A: Left panel accounts for Age, BMI, DCAL (diet) and number of trainings in
    the estimation and test of the differential expression of every gene; it
    may present marginal differences with the version presented 2 years ago
    likely due to slight variations in stochastic elements (e.g. missing
    data imputation).",
    "A: Right panel discards any confounding factors.",
    "B, C: Scatter plots comparing
    the Log Fold Change estimations (B)/
    the significance (C, the higher the more significant)
    with (y axis) without (x axis) adjustment for confounding factors with
    annotation corresponding to the Pearson's correlation (r) and
    Spearman's rank correlation (rho).",
    "Hexbin representation: the intensity of each hexagonal bin accounts for
    the number of genes found in the area it covers."
  )) |>
  add_figure(p, height = .9) |>
  add_footer(
    "Program t2d_06_rna / Env ayup_dbs:v0.1.0-alpha",
    params$version
  )
```

```
Warning: Removed 1 rows containing missing values (geom_text).
```

```
export_as(
  p,
  file = file.path(params$paths$grh, "fig_06_01.pdf"),
  file_graph_alone = file.path(params$paths$grh, "fig_06_01_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_06_01.pdf
```

```
[log] output saved as: ../tlg/graph/fig_06_01_af.pdf (annot. free)
```

```
show_slate(p)
```

4.4 Session Informations

```
sessioninfo::session_info()
```

5 mRNA / micro RNA

Program 07

Francois Collin
2022-12-12

```
cfg_prog <- yaml::read_yaml("_prog.yml")
```

The variation in messenger RNA (mRNA) and micro-RNA (miRNA) abundance measured by RNA-Seq associated either with the dysglycemia and/or the 3-month exercise intervention were investigated by a Differential Expression analysis (DE). At baseline, the DE models included only the effect of the dysglycemia-constrast group, consistently with the Anova model applied for demographics and baseline anthropometrics. To do so, genetic features which were never found expressed were discarded. Then RNA-Seq libraries were included if reaching a read depth threshold (total number of reads per sample) fixed by visual examination of the association between sample gene diversity and sample depth. For the remaining libraries, genetic features were included if the count per million reads (CPM) was greater than 2 in at least 50% of the samples. The DE analysis implementation proposed by Love, Huber, and Anders (2014) with the R package `DESeq2` was used to fit the models; it is based on a negative binomial distribution accounting for read variability, dispersion corrected after trends seen across all samples and genes (Dündar, Skrabanek, and Zumbo 2018). In addition, the number of tested genes being high, raw p.values were adjusted according to Benjamini and Hochberg's method also known as False-Discovery Rate (FDR).

The analysis was realized with R (v4.2.0, R Core Team 2022). Note the use of the additional packages downloaded from the RStudio package manager repository (freeze date 2022-05-04, repos):

- `DESeq2` for differential expression analysis and variance stabilizing transformation (version 1.36.0, Love, Huber, and Anders 2014)
- `ggplot2` for graphics (version 3.3.6, Wickham 2016)
- `MultiAssayExperiment` for management of multi-omics experiment objects (version 1.22.0, Ramos et al. 2017).

Target:

- ☒ miRNA DE at baseline without confounding factors.
- ☒ mRNA DE at baseline without confounding factors.
- ☒ miRNA DE at month 3 without confounding factors.
- ☒ mRNA DE at month 3 without confounding factors.
- ☒ miRNA DE variation induced by the exercise intervention.
- ☒ mRNA DE variation induced by the exercise intervention.

```
devtools::load_all("src/pkg/dbs.data")
```

```
i Loading dbs.data
```

```
devtools::load_all("src/pkg/latarnia.utils")
```

```
i Loading latarnia.utils
Loading required package: grid

Loading required package: shiny
```

```
knitr::opts_chunk$set(results = cfg_prog$knitr$results)
library(assertthat)
library(ggplot2)
source("R/ngs.R")
source("R/inches.R")
source("R/export_xl.R")
```

5.0.1 Materials and Methods

5.0.1.1 Helper functions

```
# Help for data pre-processing
filter_for_depth <- function(mae, assay, min_depth) {
  mae[, colSums(mae[[ngs_assay]]) > min_depth, ]
}

filter_for_coldata <- function(mae, col = "avisit", is) {
  mae[, colData(mae)[[col]] %in% is, ]
}
```

```
filter_for_low_expr <- function(mae, assay, min_cpm,
                                frac_cols = cfg_prog$rna$cpm_threshold$fraccol
) {
  # Genes expressed at least cpm_threshold in frac_cols columns
  mae[
    rowSums(cpm(mae[[assay]]) > min_cpm) >
      ncol(mae[[assay]]) * frac_cols,
    ,
  ]
}

# Differential expression helper functions:
de_by_ctrs <- function(df,
                       ctrs,
                       adj_meth = ctrl$adj_meth) {
  lapply(
    ctrs,
    fit = df,
    adj_meth = adj_meth,
    FUN = function(x, fit, adj_meth) {
      y <- DESeq2::results(fit, contrast = x, pAdjustMethod = adj_meth)
      y$feature <- rownames(y)
      y$log_padj <- -1 * log10(y$padj)
      y$ctrs <- paste(x["test"], "vs", x["ref"])
      as.data.frame(y)
    }
  )
}
```

5.0.1.2 Data preparation

```
adsl <- dbs.data::adsl
advs <- dbs.data::advs
adlb <- dbs.data::adlb

#' Subjid and Visit to Sample
#'
subjvis_to_spl <- function(df) paste0(df$subjid, "v", df$avisitn)

ads <- adlb |>
```

```
  rbind(advs) |>
  subset(select = -c(ct, dtype, param, base, basetype, chg, pchg)) |>
  tidyr::pivot_wider(names_from = "paramcd", values_from = "aval") |>
  merge(adsl, y = _, by = "subjid") |>
  (\(df) S4Vectors::DataFrame(df, row.names = subjvis_to_spl(df)))() |>
  subset(select = c(subjid, diabcd, diab, avisitn, avisit)) |>
  (\(df) {
    assertthat::assert_that(all(table(subjvis_to_spl(df)) == 1))
    df
  })()

rm(adsl, advs, adlb)
ads
```

```
rna <- list(
  mrna = dbs.data::mrna_raw,
  premirna = dbs.data::premirna_raw,
  mirna = dbs.data::mirna_raw
)

rna[c("premirna", "mirna")] <- lapply(
  X = rna[c("premirna", "mirna")],
  FUN = format_mirna
)

# Rows represent genes
rna <- lapply(X = rna, FUN = function(x) y <- x[rowSums(x) > 0, ])
rna <- lapply(X = rna, as.matrix)
assertthat::assert_that(all(colnames(rna$premirna) == colnames(rna$mirna)))
rna$allmirna <- rbind(rna$premirna, rna$mirna)

library(testthat)
test_that("rna features discriminated in noexpr, expr", {
  lapply(X = rna, FUN = \(x) expect_true(all(rowSums(x) > 0)))
})
```

```
library(MultiAssayExperiment)
```

```
Loading required package: SummarizedExperiment
```

```
Loading required package: MatrixGenerics
```

```
Loading required package: matrixStats


Attaching package: 'MatrixGenerics'

The following objects are masked from 'package:matrixStats':

    colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
    colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
    colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
    colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
    colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
    colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
    colWeightedMeans, colWeightedMedians, colWeightedSds,
    colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
    rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
    rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
    rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
    rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
    rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
    rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
    rowWeightedSds, rowWeightedVars

Loading required package: GenomicRanges

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, 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, setdiff, sort, table, tapply,
    union, unique, unsplit, which.max, which.min

Loading required package: S4Vectors


Attaching package: 'S4Vectors'

The following objects are masked from 'package:base':

    expand.grid, I, unname

Loading required package: IRanges

Loading required package: GenomeInfoDb

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")'.


Attaching package: 'Biobase'

The following object is masked from 'package:MatrixGenerics':

    rowMedians

The following objects are masked from 'package:matrixStats':

    anyMissing, rowMedians
```

```r
#' (Sample-)Map Arrays
#'
#' Use the column names of `x` to deduce the `primary` and `colnames`.
#' This is used to generate the sample mapping between colData and Experiments.
#'
#' @param x (`dataframe`).
#'
#' @note In our case, primary and colnames are equivalent, colnames could
#' be different from primary names when a biological sample has different
#' names in the biological assays (e.g. machine constraint, technical
#' repetitions).
#'
#' @seealso [MultiAssayExperiment::listToMap()]
#' @examples
#' \dontrun{
#'   lapply(rna, map_arrays)
#'   MultiAssayExperiment::listToMap(lapply(rna, map_arrays))
#' }
#'
map_arrays <- function(x) {
  y <- data.frame(colname = colnames(x))
  y$primary <- y$colname
  y
}

besd_mae <- MultiAssayExperiment(
  experiments = ExperimentList(rna),
  colData = ads,
  sampleMap = listToMap(lapply(rna, map_arrays))
)

rm(rna, ads)
besd_mae
```

5.1 Sample depth threshold determination

A scatter plot representing sample gene diversity in relationship with sample depth evidenced a threshold for miRNA reads at 5e+06 reads for which approximately a million read increase was associated with more than a 2-fold increase in gene diversity. The depth threshold for mRNA was fixed at , although with a less important but still substantial drop in gene variety

below that threshold.

5.1.1 miRNA

```
assay <- "allmirna"
```

5.1.1.1 Static figure

```
vline <- cfg_prog$rna$depth_threshold[[assay]]
dtaplot <- assays(besd_mae)[[assay]]
dtaplot <- data.frame(
  n_gene = colSums(dtaplot > 0),
  depth  = colSums(dtaplot)
)

dtaplot <- cbind(
  dtaplot,
  data.frame(colData(besd_mae)[rownames(dtaplot),])
)

gg <- ggplot(
  data = dtaplot,
  mapping = aes(x = depth, y = n_gene, colour = diab)
) +
  geom_point() +
  geom_vline(xintercept = vline) +
  xlab("Sample depth") +
  ylab("Gene diversity (number of genes with reads)") +
  coord_cartesian(ylim = c(0, NA)) +
  theme_minimal() +
  theme(legend.position = "top")

gg
```

5.1.1.2 Interactive figure

```
plotly::ggplotly(gg)
```

```
rm(vline, dtaplot, assay, gg)
```

```
Warning in rm(vline, dtaplot, assay, gg): object 'vline' not found
```

```
Warning in rm(vline, dtaplot, assay, gg): object 'dtaplot' not found
```

```
Warning in rm(vline, dtaplot, assay, gg): object 'gg' not found
```

5.1.2 mRNA

5.1.2.1 Static figure

```
assay <- "mrna"
```

```
vline <- cfg_prog$rna$depth_threshold[[assay]]
dtaplot <- assays(besd_mae)[[assay]]
dtaplot <- data.frame(
  n_gene = colSums(dtaplot > 0),
  depth  = colSums(dtaplot)
)

dtaplot <- cbind(
  dtaplot,
  data.frame(colData(besd_mae)[rownames(dtaplot),])
)

gg <- ggplot(
  data = dtaplot,
  mapping = aes(x = depth, y = n_gene, colour = diab)
) +
  geom_point() +
  geom_vline(xintercept = vline) +
  xlab("Sample depth") +
  ylab("Gene diversity (number of genes with reads)") +
```

```
  coord_cartesian(ylim = c(0, NA)) +
  theme_minimal() +
  theme(legend.position = "top")

gg
```

5.1.2.2 Interactive figure

```
plotly::ggplotly(gg)
```

:::

5.2 `dds_1` - DE: Baseline, all micro RNA

```
ctrl <- cfg_prog$rna
ngs_assay <- "allmirna"

dds_1_fit <- besd_mae |>
  (\(mae) mae[ , , ngs_assay])() |>
  filter_for_depth(
    assay = ngs_assay,
    min_depth = ctrl$depth_threshold[[ngs_assay]]
  ) |>
  filter_for_coldata(col = "avisit", is = "Baseline") |>
  filter_for_low_expr(
    assay = ngs_assay,
    min_cpm = ctrl$cpm_threshold[[ngs_assay]],
    frac_cols = ctrl$cpm_threshold$fraccol
  ) |>
  (\(mae) DESeq2::DESeqDataSetFromMatrix(
    countData = mae[[ngs_assay]],
    colData = colData(x = mae),
    design = stats::formula(~ diabcd)
  ) )() |>
  DESeq2::DESeq(
    parallel = TRUE,
    BPPARAM = BiocParallel::bpparam(),
    fitType = "local"
```

```
    )
```

```
Warning: 'experiments' dropped; see 'metadata'
```

```
harmonizing input:
  removing 282 sampleMap rows not in names(experiments)
```

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates: 2 workers
```

```
mean-dispersion relationship
```

```
final dispersion estimates, fitting model and testing: 2 workers
```

```
-- replacing outliers and refitting for 13 genes
-- DESeq argument 'minReplicatesForReplace' = 7
-- original counts are preserved in counts(dds)
```

```
estimating dispersions
```

```
fitting model and testing
```

```
dds_1_est <- dds_1_fit |>
  de_by_ctrs(
    ctrs = list(
      c("diabcd", test = "T2D", ref = "NGT"),
      c("diabcd", test = "T2D", ref = "IGT"),
      c("diabcd", test = "IGT", ref = "NGT")
    )
  ) |>
  Reduce(rbind, x = _) |>
  within({
    label_xp(lfcSE) <- "Log Fold Change Standard Error"
    label_xp(ctrs) <- "Contrast"
```

```
    label_xp(padj) <- "Adjusted p.value (False-Discovery Rate)"
    label_xp(log_padj) <- "-1 * log10(padj)"
  })
```

5.2.1 Fig 07 01

```
pval <- c(0.05, 0.001)
log_adj <- pretty(dds_1_est$log_padj)
brk <- setNames(
  c(log_adj, -1 * log10(pval)),
  c(log_adj, paste0("padj=", pval))
)
dds_1_gg <-
  dds_1_est |>
  ggplot(aes(log2FoldChange, log_padj, fill = log10(..count..))) +
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
  stat_bin_hex() +
  scale_fill_gradient(low = "black", high = "gray90") +
  scale_y_continuous(breaks = brk) +
  facet_grid(. ~ ctrs) +
  theme_minimal() +
  theme(
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.position = "bottom",
    legend.key.height = unit(.5, "lines"),
    legend.key.width = unit(3, "lines"),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )
```

```
p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 7.1",
      strwrap(
```

```
        "Volcano plot - Response Size and Significance of Differential
        Expression among all miRNA at Baseline", width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(dds_1_gg, height = inches(3), width = inches(6)) |>
  add_footer(
    c(
      "Program t2d_07_mir / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )
```

```
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, "fig_07_01.pdf"),
  file_graph_alone = file.path(cfg_prog$paths$grh, "fig_07_01_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_07_01.pdf
```

```
[log] output saved as: ../tlg/graph/fig_07_01_af.pdf (annot. free)
```

```
show_slate(p)
```

```
export_xl(
  fig_07_01 = dds_1_est,
  info = c(
    title =
      "miRNA at baseline - differential expression by diabetes group",
    source = "UMB-BESD"
  ),
  dir = cfg_prog$paths$grh,
  basenm = "fig_07_01"
)
```

```
Excel output: ../tlg/graph/fig_07_01.xlsx
```

5.2.2 Fig 07 02

```
pval <- c(0.05, 0.001)
log_adj <- pretty(dds_1_est$log_padj)
brk <- setNames(
  c(log_adj, -1 * log10(pval)),
  c(log_adj, paste0("padj=", pval))
)

dds_1_est$label <- gsub(
  "hsa-(mir|miR|let)-",
  dds_1_est$feature,
  replacement = ""
)
dds_1_est$type <- substr(dds_1_est$feature, start = 5, stop = 7)
head(dds_1_est)

dds_1_gg <- ggplot(
  dds_1_est,
  mapping = aes(x = log2FoldChange, y = log_padj, col = type)
) +
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
  stat_bin_hex(mapping = aes(fill = log10(..count..)), color = "transparent") +
  geom_point(
    data = subset(dds_1_est, padj < 0.001 & abs(log2FoldChange) > 1),
    size = 1,
    shape = 3
  ) +
  geom_text(
    data = subset(dds_1_est, padj < 0.001 & abs(log2FoldChange) > 1),
    aes(label = label),
    vjust = -0.5,
    size = 2,
    show.legend = FALSE
  ) +
  scale_fill_gradient(low = "black", high = "gray90") +
  scale_color_manual(values = c("#FF5003", "#003F7D")) +
  scale_y_continuous(breaks = brk) +
  facet_wrap(. ~ ctrs, ncol = 3) +
```

```
  coord_cartesian(clip = "off") +
  theme_minimal() +
  theme(
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.position = "bottom",
    legend.key.height = unit(.5, "lines"),
    legend.key.width = unit(2, "lines"),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )

p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 7.2",
      strwrap(
        "Volcano plot - Response Size and Significance of Differential
        Expression among all miRNA at Baseline", width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(dds_1_gg, height = inches(3), width = inches(6)) |>
  add_footer(
    c(
      "Program t2d_07_mir / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )

export_as(
  p,
  file = file.path(cfg_prog$paths$grh, "fig_07_02.pdf"),
  file_graph_alone = file.path(cfg_prog$paths$grh, "fig_07_02_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_07_02.pdf

[log] output saved as: ../tlg/graph/fig_07_02_af.pdf (annot. free)
```

```
show_slate(p)
```

```
rm(ctrl, ngs_assay, dds_1_fit, pval, log_adj, brk, dds_1_gg, p)
gc()
ls()
```

5.3 `dds_2` - DE: Baseline, mRNA

```
ctrl <- cfg_prog$rna
ngs_assay <- "mrna"

dds_2_fit <- besd_mae |>
  (\(mae) mae[ , , ngs_assay])() |>
  filter_for_depth(
    assay = ngs_assay,
    min_depth = ctrl$depth_threshold[[ngs_assay]]
  ) |>
  filter_for_coldata(col = "avisit", is = "Baseline") |>
  filter_for_low_expr(
    assay = ngs_assay,
    min_cpm = ctrl$cpm_threshold[[ngs_assay]]
  ) |>
  (\(mae) DESeq2::DESeqDataSetFromMatrix(
    countData = mae[[ngs_assay]],
    colData = colData(x = mae),
    design = stats::formula(~ diabcd)
  ) )() |>
  DESeq2::DESeq(
    parallel = TRUE,
    BPPARAM = BiocParallel::bpparam()
  )
```

```
Warning: 'experiments' dropped; see 'metadata'
```

```
harmonizing input:
  removing 282 sampleMap rows not in names(experiments)
```

```
estimating size factors
```

```
estimating dispersions

gene-wise dispersion estimates: 2 workers

mean-dispersion relationship

final dispersion estimates, fitting model and testing: 2 workers

-- replacing outliers and refitting for 13 genes
-- DESeq argument 'minReplicatesForReplace' = 7
-- original counts are preserved in counts(dds)

estimating dispersions

fitting model and testing
```

```
dds_2_est <- dds_2_fit |>
  de_by_ctrs(
    ctrs = list(
      c("diabcd", test = "T2D", ref = "NGT"),
      c("diabcd", test = "T2D", ref = "IGT"),
      c("diabcd", test = "IGT", ref = "NGT")
    )
  ) |>
  Reduce(rbind, x = _) |>
  within({
    label_xp(lfcSE) <- "Log Fold Change Standard Error"
    label_xp(ctrs) <- "Contrast"
    label_xp(padj) <- "Adjusted p.value (False-Discovery Rate)"
    label_xp(log_padj) <- "-1 * log10(padj)"
  })
```

```
head(dds_2_est)
```

```
ndegenes <- dds_2_est |>
  subset(round(padj, 5) < 0.001) |>
  with(unique(feature)) |>
  length()
```

```r
sprintf("%.2f", round(ndegenes / ngenes * 100, 2))

split(dds_2_est, dds_2_est$ctrs) |>
  lapply(\(x) {
    dim(x)
    head(x)
    table(cut(x$padj, breaks = c(0, 0.001, 0.05, 1)))
  })



pct_de_genes <- sprintf("%.2f", round(ndegenes / ngenes * 100, 2))


genes_nsub <- nrow(besd_mae[[ngs_assay]])
genes_de_n <- c(
  Overall = dds_2_est |>
      subset(round(padj, 5) < 0.001) |>
      with(unique(feature)) |>
      length()
  ,
  split(dds_2_est, dds_2_est$ctrs) |>
    sapply(\(x) {
      x <- subset(x, round(padj, 5) < 0.001)
      length(unique(x$feature))
    })
)

genes_nsub
genes_de_n
genes_de_pct <- sprintf("%.2f", genes_de_n / genes_nsub * 100)
```

5.3.1 Fig 07 03

```r
pval <- c(0.05, 0.001)
log_adj <- pretty(dds_2_est$log_padj)
brk <- setNames(
  c(log_adj, -1 * log10(pval)),
```

```
  c(log_adj, paste0("padj=", pval))
)

dds_2_gg <-
  dds_2_est |>
  ggplot(aes(log2FoldChange, log_padj, fill = log10(..count..))) +
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
  stat_bin_hex() +
  scale_fill_gradient(low = "black", high = "gray90") +
  scale_y_continuous(breaks = brk) +
  facet_grid(. ~ ctrs) +
  theme_minimal() +
  theme(
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.position = "bottom",
    legend.key.height = unit(.5, "lines"),
    legend.key.width = unit(3, "lines"),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )
```

```
p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 7.3",
      strwrap(
        "Volcano plot - Response Size and Significance of Differential
        Expression among mRNA at Baseline", width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(dds_2_gg, height = inches(3), width = inches(6)) |>
  add_footer(
    c(
      "Program t2d_07_mir / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
```

```
    ),
    cfg_prog$version
  )
```

```
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, "fig_07_03.pdf"),
  file_graph_alone = file.path(cfg_prog$paths$grh, "fig_07_03_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_07_03.pdf
```

```
[log] output saved as: ../tlg/graph/fig_07_03_af.pdf (annot. free)
```

```
show_slate(p)
```

```
export_xl(
  fig_07_03 = dds_2_est,
  info = c(
    title =
      "mRNA at baseline - differential expression by diabetes group",
    source = "UMB-BESD"
  ),
  dir = cfg_prog$paths$grh,
  basenm = "fig_07_03"
)
```

```
Excel output: ../tlg/graph/fig_07_03.xlsx
```

5.3.2 Fig 07 04

```
dds_2_est$label <- gsub("ENSG0+", "", dds_2_est$feature)
head(dds_2_est)

pval <- c(0.05, 0.001)
log_adj <- pretty(dds_2_est$log_padj)
brk <- setNames(
  c(log_adj, -1 * log10(pval)),
```

```
  c(log_adj, paste0("padj=", pval))
)

dds_2_gg <- ggplot(dds_2_est, mapping = aes(x = log2FoldChange, y = log_padj)) +
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
  stat_bin_hex(mapping = aes(fill = log10(..count..)), color = "transparent") +
  geom_point(
    data = subset(dds_2_est, padj < 0.001 & abs(log2FoldChange) > 2),
    size = 1,
    shape = 3,
    color = "royalblue"
  ) +
  geom_text(
    data = subset(dds_2_est, padj < 0.001 & abs(log2FoldChange) > 2),
    aes(label = label),
    vjust = -0.5,
    size = 2,
    color = "royalblue",
    show.legend = FALSE
  ) +
  scale_fill_gradient(low = "black", high = "gray90") +
  scale_color_manual(values = c("#FF5003", "#003F7D")) +
  scale_y_continuous(breaks = brk) +
  facet_wrap(. ~ ctrs, ncol = 3) +
  coord_cartesian(clip = "off") +
  theme_minimal() +
  theme(
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.position = "bottom",
    legend.key.height = unit(.5, "lines"),
    legend.key.width = unit(2, "lines"),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )

p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
```

```
    c(
      "Figure 7.4",
      strwrap(
        "Volcano plot - Response Size and Significance of Differential
        Expression among all miRNA at Baseline", width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(dds_2_gg, height = inches(3), width = inches(6)) |>
  add_footer(
    c(
      "Program t2d_07_mir / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )

export_as(
  p,
  file = file.path(cfg_prog$paths$grh, "fig_07_04.pdf"),
  file_graph_alone = file.path(cfg_prog$paths$grh, "fig_07_04_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_07_04.pdf
```

```
[log] output saved as: ../tlg/graph/fig_07_04_af.pdf (annot. free)
```

```
show_slate(p)
```

5.4 dds_3 - DE: Month 3, all micro RNA

```
ctrl <- cfg_prog$rna
ngs_assay <- "allmirna"

dds_3_fit <- besd_mae |>
  (\(mae) mae[ , , ngs_assay])() |>
  filter_for_depth(
```

```
    assay = ngs_assay,
    min_depth = ctrl$depth_threshold[[ngs_assay]]
  ) |>
  filter_for_coldata(col = "avisit", is = "Month 3") |>
  filter_for_low_expr(
    assay = ngs_assay,
    min_cpm = ctrl$cpm_threshold[[ngs_assay]]
  ) |>
  (\(mae) DESeq2::DESeqDataSetFromMatrix(
    countData = mae[[ngs_assay]],
    colData = colData(x = mae),
    design = stats::formula(~ diabcd)
  ) )() |>
  DESeq2::DESeq(
    parallel = TRUE,
    BPPARAM = BiocParallel::bpparam(),
    fitType = "local"
  )
```

```
Warning: 'experiments' dropped; see 'metadata'

harmonizing input:
  removing 282 sampleMap rows not in names(experiments)

estimating size factors

estimating dispersions

gene-wise dispersion estimates: 2 workers

mean-dispersion relationship

final dispersion estimates, fitting model and testing: 2 workers

-- replacing outliers and refitting for 13 genes
-- DESeq argument 'minReplicatesForReplace' = 7
-- original counts are preserved in counts(dds)

estimating dispersions

fitting model and testing
```

```
dds_3_est <-
  dds_3_fit |>
  de_by_ctrs(
    ctrs = list(
      c("diabcd", test = "T2D", ref = "NGT"),
      c("diabcd", test = "T2D", ref = "IGT"),
      c("diabcd", test = "IGT", ref = "NGT")
    )
  ) |>
  Reduce(rbind, x = _) |>
  within({
    label_xp(lfcSE) <- "Log Fold Change Standard Error"
    label_xp(ctrs) <- "Contrast"
    label_xp(padj) <- "Adjusted p.value (False-Discovery Rate)"
    label_xp(log_padj) <- "-1 * log10(padj)"
  })
```

5.4.1 Fig 07 05

```
pval <- c(0.05, 0.001)
log_adj <- pretty(dds_3_est$log_padj)
brk <- setNames(
  c(log_adj, -1 * log10(pval)),
  c(log_adj, paste0("padj=", pval))
)
dds_3_gg <-
  dds_3_est |>
  ggplot(aes(log2FoldChange, log_padj, fill = log10(..count..))) +
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
  stat_bin_hex() +
  scale_fill_gradient(low = "black", high = "gray90") +
  scale_y_continuous(breaks = brk) +
  facet_grid(. ~ ctrs) +
  theme_minimal() +
  theme(
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.position = "bottom",
```

```
    legend.key.height = unit(.5, "lines"),
    legend.key.width = unit(3, "lines"),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )

p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 7.5",
      strwrap(
        "Volcano plot - Response Size and Significance of Differential
        Expression among all miRNA at Month 3", width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(dds_3_gg, height = inches(3), width = inches(6)) |>
  add_footer(
    c(
      "Program t2d_07_mir / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )
```

```
Warning: Removed 350 rows containing non-finite values (stat_binhex).
```

```
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, "fig_07_05.pdf"),
  file_graph_alone = file.path(cfg_prog$paths$grh, "fig_07_05_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_07_05.pdf
```

```
[log] output saved as: ../tlg/graph/fig_07_05_af.pdf (annot. free)
```

```
show_slate(p)
```

```
export_xl(
  fig_07_05 = dds_3_est,
  info = c(
    title =
      "All miRNA at month 3 - differential expression by diabetes group",
    source = "UMB-BESD"
  ),
  dir = cfg_prog$paths$grh,
  basenm = "fig_07_05"
)
```

Excel output: ../tlg/graph/fig_07_05.xlsx

5.4.2 Fig 07 06

```
pval <- c(0.05, 0.001)
log_adj <- pretty(dds_3_est$log_padj)
brk <- setNames(
  c(log_adj, -1 * log10(pval)),
  c(log_adj, paste0("padj=", pval))
)

dds_3_est$label <- gsub(
  "hsa-(mir|miR|let)-",
  dds_3_est$feature,
  replacement = ""
)
dds_3_est$type <- substr(dds_3_est$feature, start = 5, stop = 7)
head(dds_3_est)

dds_3_gg <- ggplot(
  dds_3_est,
  mapping = aes(x = log2FoldChange, y = log_padj, col = type)
) +
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
```

```
  stat_bin_hex(mapping = aes(fill = log10(..count..)), color = "transparent") +
  geom_point(
    data = subset(dds_3_est, padj < 0.001 & abs(log2FoldChange) > 1),
    size = 1,
    shape = 3
  ) +
  geom_text(
    data = subset(dds_3_est, padj < 0.001 & abs(log2FoldChange) > 1),
    aes(label = label),
    vjust = -0.5,
    size = 2,
    show.legend = FALSE
  ) +
  scale_fill_gradient(low = "black", high = "gray90") +
  scale_color_manual(values = c("#FF5003", "#003F7D")) +
  scale_y_continuous(breaks = brk) +
  facet_wrap(. ~ ctrs, ncol = 3) +
  coord_cartesian(clip = "off") +
  theme_minimal() +
  theme(
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.position = "bottom",
    legend.key.height = unit(.5, "lines"),
    legend.key.width = unit(2, "lines"),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )

p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 7.6",
      strwrap(
        "Volcano plot - Response Size and Significance of Differential
        Expression among all miRNA at Month 3", width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
```

```
  add_figure(dds_3_gg, height = inches(3), width = inches(6)) |>
  add_footer(
    c(
      "Program t2d_07_mir / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )
```

```
Warning: Removed 350 rows containing non-finite values (stat_binhex).
```

```
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, "fig_07_06.pdf"),
  file_graph_alone = file.path(cfg_prog$paths$grh, "fig_07_06_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_07_06.pdf
```

```
[log] output saved as: ../tlg/graph/fig_07_06_af.pdf (annot. free)
```

```
show_slate(p)
```

5.5 dds_4 - DE: Month 3, mRNA

```
ctrl <- cfg_prog$rna
ngs_assay <- "mrna"

dds_4_fit <- besd_mae |>
  (\(mae) mae[ , , ngs_assay])() |>
  filter_for_depth(
    assay = ngs_assay,
    min_depth = ctrl$depth_threshold[[ngs_assay]]
  ) |>
  filter_for_coldata(col = "avisit", is = "Month 3") |>
  filter_for_low_expr(
    assay = ngs_assay,
```

```
    min_cpm = ctrl$cpm_threshold[[ngs_assay]]
  ) |>
  (\(mae) DESeq2::DESeqDataSetFromMatrix(
    countData = mae[[ngs_assay]],
    colData = colData(x = mae),
    design = stats::formula(~ diabcd)
  ) )() |>
  DESeq2::DESeq(
    parallel = TRUE,
    BPPARAM = BiocParallel::bpparam()
  )
```

```
Warning: 'experiments' dropped; see 'metadata'
```

```
harmonizing input:
  removing 282 sampleMap rows not in names(experiments)
```

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates: 2 workers
```

```
mean-dispersion relationship
```

```
final dispersion estimates, fitting model and testing: 2 workers
```

```
-- replacing outliers and refitting for 21 genes
-- DESeq argument 'minReplicatesForReplace' = 7
-- original counts are preserved in counts(dds)
```

```
estimating dispersions
```

```
fitting model and testing
```

```
dds_4_est <-
  dds_4_fit |>
```

```
  de_by_ctrs(
    ctrs = list(
      c("diabcd", test = "T2D", ref = "NGT"),
      c("diabcd", test = "T2D", ref = "IGT"),
      c("diabcd", test = "IGT", ref = "NGT")
    )
  ) |>
  Reduce(rbind, x = _) |>
  within({
    label_xp(lfcSE) <- "Log Fold Change Standard Error"
    label_xp(ctrs) <- "Contrast"
    label_xp(padj) <- "Adjusted p.value (False-Discovery Rate)"
    label_xp(log_padj) <- "-1 * log10(padj)"
  })
```

5.5.1 Fig 07 07

```
pval <- c(0.05, 0.001)
log_adj <- pretty(dds_4_est$log_padj)
brk <- setNames(
  c(log_adj, -1 * log10(pval)),
  c(log_adj, paste0("padj=", pval))
)

dds_4_gg <- dds_4_est |>
  # All pval close to 1 disturbe the hex-density plots
  (\(df) {
    assert_that(all(df$padj[df$ctrs == "IGT vs NGT"] > 0.991))
    df
  })() |>
  subset(ctrs != "IGT vs NGT") |>
  ggplot(aes(log2FoldChange, log_padj, fill = log10(..count..))) +
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
  stat_bin_hex() +
  scale_fill_gradient(low = "black", high = "gray90") +
  scale_y_continuous(breaks = brk) +
  facet_grid(. ~ ctrs) +
  theme_minimal() +
```

```
  theme(
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.position = "bottom",
    legend.key.height = unit(.5, "lines"),
    legend.key.width = unit(3, "lines"),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )

p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 7.7",
      strwrap(
        "Volcano plot - Response Size and Significance of Differential
        Expression among all mRNA at Month 3", width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(dds_4_gg, height = inches(3), width = inches(6)) |>
  add_note(c(
    "NOTE: all feature comparisons between IGT and NGT were characterised by
    adjusted p.values greater than 0.99; to preserve the hexagonal-density
    plot, this comparison was not represented."
  )) |>
  add_footer(
    c(
      "Program t2d_07_mir / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )

export_as(
  p,
  file = file.path(cfg_prog$paths$grh, "fig_07_07.pdf"),
  file_graph_alone = file.path(cfg_prog$paths$grh, "fig_07_07_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_07_07.pdf

[log] output saved as: ../tlg/graph/fig_07_07_af.pdf (annot. free)
```

```
show_slate(p)
```

```
export_xl(
  fig_07_07 = dds_4_est,
  info = c(
    title =
      "mRNA at month 3 - differential expression by diabetes group",
    source = "UMB-BESD"
  ),
  dir = cfg_prog$paths$grh,
  basenm = "fig_07_07"
)
```

```
Excel output: ../tlg/graph/fig_07_07.xlsx
```

5.5.2 Fig 07 08

```
dds_4_est$label <- gsub("ENSG0+", "", dds_4_est$feature)
head(dds_4_est)

pval <- c(0.05, 0.001)
log_adj <- pretty(dds_4_est$log_padj)
brk <- setNames(
  c(log_adj, -1 * log10(pval)),
  c(log_adj, paste0("padj=", pval))
)

dds_4_gg <- dds_4_est |>
  # All pval close to 1 disturbe the hex-density plots
  (\(df) {
    assert_that(all(df$padj[df$ctrs == "IGT vs NGT"] > 0.991))
    df
  })() |>
  subset(ctrs != "IGT vs NGT") |>
  ggplot(mapping = aes(x = log2FoldChange, y = log_padj)) +
```

```
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
  stat_bin_hex(mapping = aes(fill = log10(..count..)), color = "transparent") +
  geom_point(
    data = subset(dds_4_est, padj < 0.001 & abs(log2FoldChange) > 2),
    size = 1,
    shape = 3,
    color = "royalblue"
  ) +
  geom_text(
    data = subset(dds_4_est, padj < 0.001 & abs(log2FoldChange) > 2),
    aes(label = label),
    vjust = -0.5,
    size = 2,
    color = "royalblue",
    show.legend = FALSE
  ) +
  scale_fill_gradient(low = "black", high = "gray90") +
  scale_color_manual(values = c("#FF5003", "#003F7D")) +
  scale_y_continuous(breaks = brk) +
  facet_wrap(. ~ ctrs, ncol = 3) +
  coord_cartesian(clip = "off") +
  theme_minimal() +
  theme(
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.position = "bottom",
    legend.key.height = unit(.5, "lines"),
    legend.key.width = unit(2, "lines"),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )

p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 7.8",
      strwrap(
        "Volcano plot - Response Size and Significance of Differential
```

```
        Expression among mRNA at Month 3", width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(dds_4_gg, height = inches(3), width = inches(6)) |>
  add_note(c(
    "NOTE: all feature comparisons between IGT and NGT were characterised by
    adjusted p.values greater than 0.99; to preserve the hexagonal-density
    plot, this comparison was not represented."
  )) |>
  add_footer(
    c(
      "Program t2d_07_mir / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )

export_as(
  p,
  file = file.path(cfg_prog$paths$grh, "fig_07_08.pdf"),
  file_graph_alone = file.path(cfg_prog$paths$grh, "fig_07_08_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_07_08.pdf
```

```
[log] output saved as: ../tlg/graph/fig_07_08_af.pdf (annot. free)
```

```
show_slate(p)
```

5.6 `dds_5` - DE: Exercise intervention, all micro RNA

```
ctrl <- cfg_prog$rna
ngs_assay <- "allmirna"

filter_for_paired_visits <- function(mae) {
  assert_that(all(table(colData(mae)$subjid) <= 2))
```

```
  subj <- table(colData(mae)$subjid)
  subj <- names(subj[subj == 2])
  mae <- mae[, colData(mae)$subjid %in% subj, ]

  chk_df <- as.data.frame(colData(mae))
  check <- split(chk_df, f = chk_df$avisit)

  # Verify the selection:
  assert_that(all(
    sapply(check, subj = subj,\(x, subj) all(subj %in% x$subjid))
  ))
  mae
}

dds_5_fit <- besd_mae |>
  (\(mae) mae[ , , ngs_assay])() |>
  filter_for_depth(ngs_assay, ctrl$depth_threshold[[ngs_assay]]) |>
  filter_for_paired_visits() |>
  filter_for_low_expr(ngs_assay, ctrl$cpm_threshold[[ngs_assay]]) |>
  (\(mae) {
    # <https://support.bioconductor.org/p/84241/#84296>
    # Or:
    # vignette('DESeq2'),
    # > Section: "Group-specific condition effects, individuals nested
    # > within groups"
    colData(mae) <- droplevels(colData(mae))
    colData(mae)$subjid <- as.character(colData(mae)$subjid)
    colData(mae)$avisit_ <- factor(
      colData(mae)$avisit,
      levels = levels(colData(mae)$avisit),
      labels = sub(" ", "_", levels(colData(mae)$avisit))
    )
    coldata <- colData(mae)[c("diabcd", "subjid", "avisit_")]

    coldata <- do.call(
      rbind,
      by(
        data = coldata,
        INDICES = list(coldata$diabcd),
        FUN = function(x) {
          subjid <- as.factor(x$subjid)# nolint
```

```
            x$P.n <- factor(# nolint
              subjid,
              levels = levels(subjid),
              labels = paste0("P.", 1:nlevels(subjid))
            )
            x[order(x$subjid, x$diabcd), ]
          }
        )
      )

      coldata <- coldata[rownames(colData(mae)), ]
      dsgn <- stats::model.matrix(
        ~diabcd + diabcd:P.n + diabcd:avisit_,
        coldata
      )
      dsgn <- dsgn[, colSums(dsgn) != 0]

      DESeq2::DESeqDataSetFromMatrix(
        countData = mae[[ngs_assay]],
        colData = colData(x = mae),
        design = dsgn
      )
    })() |>
    DESeq2::DESeq(
      parallel = TRUE,
      BPPARAM = BiocParallel::bpparam(),
      fitType = "local",
      betaPrior = FALSE
    )
```

```
Warning: 'experiments' dropped; see 'metadata'
```

```
harmonizing input:
  removing 282 sampleMap rows not in names(experiments)
```

```
using supplied model matrix
```

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates: 2 workers

mean-dispersion relationship

final dispersion estimates, fitting model and testing: 2 workers
```

```
de_by_name <- function(df,
                       name,
                       adj_meth = ctrl$adj_meth) {
  lapply(
    name,
    fit = df,
    adj_meth = adj_meth,
    FUN = function(x, fit, adj_meth) {
      y <- DESeq2::results(fit, name = x, pAdjustMethod = adj_meth)
      y$feature <- rownames(y)
      y$log_padj <- -1 * log10(y$padj)
      y$ctrs <- gsub(
        pattern = "^diabcd(...)\\.avisit.*$",
        x = x,
        replacement = "chg \\1"
      )
      as.data.frame(y)
    }
  )
}

dds_5_est <-
  dds_5_fit |>
  de_by_name(
    name = c(
      chg_ngt = "diabcdNGT.avisit_Month_3",
      chg_igt = "diabcdIGT.avisit_Month_3",
      chg_t2d = "diabcdT2D.avisit_Month_3"
    )
  ) |>
  Reduce(rbind, x = _) |>
  within(
    ctrs <- factor(ctrs, levels = paste("chg", c("NGT", "IGT", "T2D")))
  ) |>
  within({
```

```
    label_xp(lfcSE) <- "Log Fold Change Standard Error"
    label_xp(ctrs) <- "Contrast"
    label_xp(padj) <- "Adjusted p.value (False-Discovery Rate)"
    label_xp(log_padj) <- "-1 * log10(padj)"
  })
```

5.6.1 Fig 07 09

```
pval <- c(0.05, 0.001)
log_adj <- pretty(dds_5_est$log_padj)
brk <- setNames(
  c(log_adj, -1 * log10(pval)),
  c(log_adj, paste0("padj=", pval))
)

dds_5_gg <-
  dds_5_est |>
  ggplot(aes(log2FoldChange, log_padj, fill = log10(..count..))) +
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
  stat_bin_hex() +
  scale_fill_gradient(low = "black", high = "gray90") +
  scale_y_continuous(breaks = brk) +
  facet_grid(. ~ ctrs) +
  theme_minimal() +
  theme(
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.position = "bottom",
    legend.key.height = unit(.5, "lines"),
    legend.key.width = unit(3, "lines"),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )

p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
```

```
      "Figure 7.9",
      strwrap(
        "Volcano plot - Size and Significance of Differential
        Expression among micro RNA in response to the exercise
        intervention (CHG)", width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(dds_5_gg, height = inches(3), width = inches(6)) |>
  add_footer(
    c(
      "Program t2d_07_mir / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )

export_as(
  p,
  file = file.path(cfg_prog$paths$grh, "fig_07_09.pdf"),
  file_graph_alone = file.path(cfg_prog$paths$grh, "fig_07_09_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_07_09.pdf
```

```
[log] output saved as: ../tlg/graph/fig_07_09_af.pdf (annot. free)
```

```
show_slate(p)
```

```
export_xl(
  fig_07_09 = dds_5_est,
  info = c(
    title = paste(
      "All miRNA Change from Baseline - differential expression by diabetes",
      "group"
    ),
    source = "UMB-BESD"
  ),
  dir = cfg_prog$paths$grh,
```

```
  basenm = "fig_07_09"
)
```

Excel output: ../tlg/graph/fig_07_09.xlsx

5.6.2 Fig 07 10

```
pval <- c(0.05, 0.001)
log_adj <- pretty(dds_5_est$log_padj)
brk <- setNames(
  c(log_adj, -1 * log10(pval)),
  c(log_adj, paste0("padj=", pval))
)

dds_5_est$label <- gsub(
  "hsa-(mir|miR|let)-",
  dds_5_est$feature,
  replacement = ""
)
dds_5_est$type <- substr(dds_5_est$feature, start = 5, stop = 7)
head(dds_5_est)

dds_5_txt <- subset(dds_5_est, padj < 0.05 & abs(log2FoldChange) > .5)

dds_5_gg <- ggplot(
  dds_5_est,
  mapping = aes(x = log2FoldChange, y = log_padj, col = type)
) +
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
  stat_bin_hex(mapping = aes(fill = log10(..count..)), color = "transparent") +
  geom_point(data = dds_5_txt, size = 1, shape = 3) +
  geom_text(
    data = dds_5_txt,
    aes(label = label),
    vjust = -0.5,
    size = 2,
    show.legend = FALSE
  ) +
```

```
  scale_fill_gradient(low = "black", high = "gray90") +
  scale_color_manual(values = c("#FF5003", "#003F7D")) +
  scale_y_continuous(breaks = brk) +
  facet_wrap(. ~ ctrs, ncol = 3) +
  coord_cartesian(clip = "off") +
  theme_minimal() +
  theme(
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.position = "bottom",
    legend.key.height = unit(.5, "lines"),
    legend.key.width = unit(2, "lines"),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )

p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 7.10",
      strwrap(
        "Volcano plot - Size and Significance of Differential
        Expression among micro RNA in response to the exercise
        intervention (CHG)", width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(dds_5_gg, height = inches(3), width = inches(6)) |>
  add_footer(
    c(
      "Program t2d_07_mir / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )

export_as(
  p,
  file = file.path(cfg_prog$paths$grh, "fig_07_10.pdf"),
```

```
  file_graph_alone = file.path(cfg_prog$paths$grh, "fig_07_10_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_07_10.pdf
```

```
[log] output saved as: ../tlg/graph/fig_07_10_af.pdf (annot. free)
```

```
show_slate(p)
```

5.7 dds_6 - DE: Exercise intervention, mRNA

```
ctrl <- cfg_prog$rna
ngs_assay <- "mrna"

filter_for_paired_visits <- function(mae) {
  assert_that(all(table(colData(mae)$subjid) <= 2))

  subj <- table(colData(mae)$subjid)
  subj <- names(subj[subj == 2])
  mae <- mae[, colData(mae)$subjid %in% subj, ]

  chk_df <- as.data.frame(colData(mae))
  check <- split(chk_df, f = chk_df$avisit)

  # Verify the selection:
  assert_that(all(
    sapply(check, subj = subj,\(x, subj) all(subj %in% x$subjid))
  ))
  mae
}

dds_6_fit <- besd_mae |>
  (\(mae) mae[ , , ngs_assay])() |>
  filter_for_depth(ngs_assay, ctrl$depth_threshold[[ngs_assay]]) |>
  filter_for_paired_visits() |>
  filter_for_low_expr(ngs_assay, ctrl$cpm_threshold[[ngs_assay]]) |>
  (\(mae) {
    # <https://support.bioconductor.org/p/84241/#84296>
```

```r
# Or:
# vignette('DESeq2'),
# > Section: "Group-specific condition effects, individuals nested
# > within groups"
colData(mae) <- droplevels(colData(mae))
colData(mae)$subjid <- as.character(colData(mae)$subjid)
colData(mae)$avisit_ <- factor(
  colData(mae)$avisit,
  levels = levels(colData(mae)$avisit),
  labels = sub(" ", "_", levels(colData(mae)$avisit))
)
coldata <- colData(mae)[c("diabcd", "subjid", "avisit_")]

coldata <- do.call(
  rbind,
  by(
    data = coldata,
    INDICES = list(coldata$diabcd),
    FUN = function(x) {
      subjid <- as.factor(x$subjid)# nolint
      x$P.n <- factor(# nolint
        subjid,
        levels = levels(subjid),
        labels = paste0("P.", 1:nlevels(subjid))
      )
      x[order(x$subjid, x$diabcd), ]
    }
  )
)

coldata <- coldata[rownames(colData(mae)), ]
dsgn <- stats::model.matrix(
  ~diabcd + diabcd:P.n + diabcd:avisit_,
  coldata
)
dsgn <- dsgn[, colSums(dsgn) != 0]

DESeq2::DESeqDataSetFromMatrix(
  countData = mae[[ngs_assay]],
  colData = colData(x = mae),
  design = dsgn
```

```
    )
  })() |>
  DESeq2::DESeq(
    parallel = TRUE,
    BPPARAM = BiocParallel::bpparam(),
    fitType = "local",
    betaPrior = FALSE
  )
```

```
Warning: 'experiments' dropped; see 'metadata'
```

```
harmonizing input:
  removing 282 sampleMap rows not in names(experiments)
```

```
using supplied model matrix
```

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates: 2 workers
```

```
mean-dispersion relationship
```

```
final dispersion estimates, fitting model and testing: 2 workers
```

```
dds_6_est <-
  dds_6_fit |>
  de_by_name(
    name = c(
      chg_ngt = "diabcdNGT.avisit_Month_3",
      chg_igt = "diabcdIGT.avisit_Month_3",
      chg_t2d = "diabcdT2D.avisit_Month_3"
    )
  ) |>
  Reduce(rbind, x = _) |>
  within(
    ctrs <- factor(ctrs, levels = paste("chg", c("NGT", "IGT", "T2D")))
```

```
  ) |>
  within({
    label_xp(lfcSE) <- "Log Fold Change Standard Error"
    label_xp(ctrs) <- "Contrast"
    label_xp(padj) <- "Adjusted p.value (False-Discovery Rate)"
    label_xp(log_padj) <- "-1 * log10(padj)"
  })
```

5.7.1 Fig 07 11

```
pval <- c(0.05, 0.001)
log_adj <- pretty(dds_6_est$log_padj)
brk <- setNames(
  c(log_adj, -1 * log10(pval)),
  c(log_adj, paste0("padj=", pval))
)

dds_6_gg <-
  dds_6_est |>
  ggplot(aes(log2FoldChange, log_padj, fill = log10(..count..))) +
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
  stat_bin_hex() +
  scale_fill_gradient(low = "black", high = "gray90") +
  scale_y_continuous(breaks = brk) +
  facet_grid(. ~ ctrs) +
  theme_minimal() +
  theme(
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.position = "bottom",
    legend.key.height = unit(.5, "lines"),
    legend.key.width = unit(3, "lines"),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )

p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
```

```
  add_title(
    c(
      "Figure 7.11",
      strwrap(
        "Volcano plot - Size and Significance of Differential
        Expression among mRNA in response to the exercise
        intervention (CHG)", width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(dds_6_gg, height = inches(3), width = inches(6)) |>
  add_footer(
    c(
      "Program t2d_07_mir / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )
```

```
Warning: Removed 1646 rows containing non-finite values (stat_binhex).
```

```
  export_as(
    p,
    file = file.path(cfg_prog$paths$grh, "fig_07_11.pdf"),
    file_graph_alone = file.path(cfg_prog$paths$grh, "fig_07_11_af.pdf")
  )
```

```
[log] output saved as: ../tlg/graph/fig_07_11.pdf
```

```
[log] output saved as: ../tlg/graph/fig_07_11_af.pdf (annot. free)
```

```
  show_slate(p)
```

```
  export_xl(
    fig_07_11 = dds_6_est,
    info = c(
      title = paste(
```

```
      "mRNA Change from Baseline - differential expression by diabetes",
      "group"
    ),
    source = "UMB-BESD"
  ),
  dir = cfg_prog$paths$grh,
  basenm = "fig_07_11"
)
```

Excel output: ../tlg/graph/fig_07_11.xlsx

5.7.2 Fig 07 12

```
pval <- c(0.05, 0.001)
log_adj <- pretty(dds_6_est$log_padj)
brk <- setNames(
  c(log_adj, -1 * log10(pval)),
  c(log_adj, paste0("padj=", pval))
)

dds_6_est$label <- gsub("ENSG0+", "", dds_6_est$feature)
dds_6_est$type <- substr(dds_6_est$feature, start = 5, stop = 7)

dds_6_txt <- subset(dds_6_est, padj < 0.05 & abs(log2FoldChange) > 1)

dds_6_gg <- ggplot(
  dds_6_est,
  mapping = aes(x = log2FoldChange, y = log_padj, col = type)
) +
  geom_vline(xintercept = c(-1, 1), lty = 2) +
  xlab("Log2-fold-change") +
  ylab(expression(-1 %*% log10(padj))) +
  stat_bin_hex(mapping = aes(fill = log10(..count..)), color = "transparent") +
  geom_point(data = dds_6_txt, size = 1, shape = 3) +
  geom_text(
    data = dds_6_txt,
    aes(label = label),
    vjust = -0.5,
    size = 2,
    show.legend = FALSE
```

```
  ) +
  scale_fill_gradient(low = "black", high = "gray90") +
  scale_color_manual(values = c("#FF5003", "#003F7D")) +
  scale_y_continuous(breaks = brk) +
  facet_wrap(. ~ ctrs, ncol = 3) +
  coord_cartesian(clip = "off") +
  theme_minimal() +
  theme(
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.position = "bottom",
    legend.key.height = unit(.5, "lines"),
    legend.key.width = unit(2, "lines"),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )

p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 7.12",
      strwrap(
        "Volcano plot - Size and Significance of Differential
        Expression among mRNA in response to the exercise
        intervention (CHG)", width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(dds_6_gg, height = inches(3), width = inches(6)) |>
  add_footer(
    c(
      "Program t2d_07_mir / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )
```

```
Warning: Removed 1646 rows containing non-finite values (stat_binhex).
```

```
  export_as(
    p,
    file = file.path(cfg_prog$paths$grh, "fig_07_12.pdf"),
    file_graph_alone = file.path(cfg_prog$paths$grh, "fig_07_12_af.pdf")
  )
```

```
[log] output saved as: ../tlg/graph/fig_07_12.pdf
```

```
[log] output saved as: ../tlg/graph/fig_07_12_af.pdf (annot. free)
```

```
  show_slate(p)
```

```
  rm(ctrl, ngs_assay, dds_6_fit, pval, log_adj, brk, dds_6_gg, p)
  gc()
  ls()
```

5.8 Session Informations

```
  sessioninfo::session_info()
```

6 WGCNA

Program 11

Francois Collin
2022-12-12

> "*The Weighted Correlation Network Analysis (WGCNA) [11,12] corresponds to a data reduction method and unsupervised classification method. It simplifies the interpretation of thousands of gene responses to a dozen of synthetic groups (or modules) of genes. The net establishes connections between genes-genes are connected if their expression is correlated. Genes can be more or less intensively connected depending on the value of the correlation (the weights). The connectivity between genes is then interpreted into a distance, and the distance is used to group genes into modules. This is how a high number of genes can be reduced into a small number of clusters whose expression is quantified by the Eigengenes (first principal component within the module). It relies on the assumption that highly correlated genes within a module are involved in common biological processes.*" Niemira et al. (2020)

Two gene co-expression networks were obtained, 1) the first one based on baseline samples for NGT and T2D subject samples and 2) the second on training-induced expression variation independently from the study arm. In detail:

1. baseline samples for NGT and T2D were selected, the same inclusion criteria as in the section "*Differential expression analysis*" were used. Gene expression was transformed using variance stabilizing transformation. The optimal soft-threshold adjacency matrix was graphically determined, then the network was estimated resulting in gene modules and corresponding Eigengenes. The `cutreeDynamic` algorithm was used for tree pruning of the gene hierarchical clustering dendrogram resulting in co-expression modules; correlated modules ($r > 0.80$) were merged. The quantitative clinical parameters were centred to the average and scaled-to-variance while factors were decomposed into dummy variables so as to study the association between gene modules and clinical traits quantified with Pearson's correlation coefficient.

2. training-induced variation was studied similarly, the approach above was applicable here also, using the difference between variance-stabilizing transformed expression matrix as input to the procedure.

Note: it is not possible to use the differentially expressed genes as a starting point. The WGCNA method includes a disclaimer preventing the use of pre-filtered / selected genes based on DEG analysis result.

The module identification was followed by gene set enrichment analysis. To do so, the correlation between gene expression (Variance-Stabilized Transformed values, VST) and the eigengene estimated in every module was used as a measure of module membership. Then, for every module, genesets as defined by the Gene Ontology (GO) project were selected if at least 10 genes belonged to the module. Then, the GSEA procedure was run while considering all expressed genes with existing *entrez gene* id and determining an enrichment score and associated p.value based on 10000 random permutations. The result focused specifically on enriched and up-regulated genesets, i.e. genesets with over representation of genes with higher correlation with the eigengene within each module.

The analysis was realized with R (v4.2.0, R Core Team 2022). Note the use of the additional packages downloaded from the RStudio package manager repository (freeze date 2022-05-04, repos):

- `MultiAssayExperiment` for management of multi-omics experiment objects (version 1.22.0, Ramos et al. 2017)
- `DESeq2` for differential expression analysis and variance stabilizing transformation (version 1.36.0, Love, Huber, and Anders 2014)
- `cowplot` for figure composition (version 1.1.1, Wilke 2020)
- `ggplot2` for graphics (version 3.3.6, Wickham 2016)
- `WGCNA` for gene clustering Langfelder and Horvath (2012).

```
cfg_prog <- yaml::read_yaml("_prog.yml")
```

```
devtools::load_all("src/pkg/dbs.data")
```

```
i Loading dbs.data
```

```
devtools::load_all("src/pkg/latarnia.utils")
```

```
i Loading latarnia.utils
Loading required package: grid

Loading required package: shiny
```

```r
knitr::opts_chunk$set(results = cfg_prog$knitr$results)

source("R/ngs.R")
source("R/inches.R")
source("R/export_xl.R")
```

6.0.1 Materials and Methods

6.0.1.1 Helper functions

```r
# Help for data pre-processing
filter_for_depth <- function(mae, assay, depth_threshold) {
  mae[, colSums(mae[[ngs_assay]]) > depth_threshold, ]
}

filter_for_visit <- function(mae, visit) {
  mae[, colData(mae)$avisit == visit, ]
}

filter_for_coldata <- function(mae, col, is) {
  mae[, colData(mae)[[col]] %in% is, ]
}

filter_for_low_expr <- function(mae, assay, cpm_threshold,
                                frac_cols = cfg_prog$rna$cpm_threshold$fraccol
) {
  # Genes expressed at least cpm_threshold in frac_cols columns
  mae[
    rowSums(cpm(mae[[assay]]) > cpm_threshold) >
      ncol(mae[[assay]]) * frac_cols,
    ,
  ]
}
```

6.0.1.2 Data preparation

```
adsl <- dbs.data::adsl
advs <- dbs.data::advs
adlb <- dbs.data::adlb

#' Subjid and Visit to Sample
#'
subjvis_to_spl <- function(df) paste0(df$subjid, "v", df$avisitn)

ads <- adlb |>
  rbind(advs) |>
  subset(select = -c(ct, dtype, param, base, basetype, chg, pchg)) |>
  tidyr::pivot_wider(names_from = "paramcd", values_from = "aval") |>
  merge(adsl, y = _, by = "subjid") |>
  (\(df) S4Vectors::DataFrame(df, row.names = subjvis_to_spl(df)))() |>
  (\(df) {
    assertthat::assert_that(all(table(subjvis_to_spl(df)) == 1))
    df
  })()
```

```
rna <- list(
  mrna = dbs.data::mrna_raw,
  premirna = dbs.data::premirna_raw,
  mirna = dbs.data::mirna_raw
)

rna[c("premirna", "mirna")] <- lapply(
  X = rna[c("premirna", "mirna")],
  FUN = format_mirna
)

# Rows represent genes
rna <- lapply(X = rna, FUN = function(x) y <- x[rowSums(x) > 0, ])
rna <- lapply(X = rna, as.matrix)
assertthat::assert_that(all(colnames(rna$premirna) == colnames(rna$mirna)))
rna$allmirna <- rbind(rna$premirna, rna$mirna)

library(testthat)
test_that("rna features discriminated in noexpr, expr", {
  lapply(X = rna, FUN = \(x) expect_true(all(rowSums(x) > 0)))
```

```
})
```

```
library(MultiAssayExperiment)
```

```
Loading required package: SummarizedExperiment
```

```
Loading required package: MatrixGenerics
```

```
Loading required package: matrixStats
```

```

Attaching package: 'MatrixGenerics'
```

```
The following objects are masked from 'package:matrixStats':

    colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
    colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
    colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
    colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
    colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
    colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
    colWeightedMeans, colWeightedMedians, colWeightedSds,
    colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
    rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
    rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
    rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
    rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
    rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
    rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
    rowWeightedSds, rowWeightedVars
```

```
Loading required package: GenomicRanges
```

```
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, 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, setdiff, sort, table, tapply,
    union, unique, unsplit, which.max, which.min

Loading required package: S4Vectors


Attaching package: 'S4Vectors'

The following objects are masked from 'package:base':

    expand.grid, I, unname

Loading required package: IRanges

Loading required package: GenomeInfoDb

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")'.


Attaching package: 'Biobase'

The following object is masked from 'package:MatrixGenerics':

    rowMedians
```

```
The following objects are masked from 'package:matrixStats':

    anyMissing, rowMedians
```

```
#' (Sample-)Map Arrays
#'
#' Use the column names of `x` to deduce the `primary` and `colnames`.
#' This is used to generate the sample mapping between colData and Experiments.
#'
#' @param x (`dataframe`).
#'
#' @note In our case, primary and colnames are equivalent, colnames could
#' be different from primary names when a biological sample has different
#' names in the biological assays (e.g. machine constraint, technical
#' repetitions).
#'
#' @seealso [MultiAssayExperiment::listToMap()]
#' @examples
#' \dontrun{
#'   lapply(rna, map_arrays)
#'   MultiAssayExperiment::listToMap(lapply(rna, map_arrays))
#' }
#'
map_arrays <- function(x) {
  y <- data.frame(colname = colnames(x))
  y$primary <- y$colname
  y
}

colnames(ads) <- tolower(colnames(ads))
besd_mae <- MultiAssayExperiment(
  experiments = ExperimentList(rna),
  colData = ads,
  sampleMap = listToMap(lapply(rna, map_arrays))
)

besd_mae
```

6.0.2 miRNA / Baseline

```
#' # https://horvath.genetics.ucla.edu/html/CoexpressionNetwork/Rpackages/WGCNA/Tutorials/
#' # WGCNA: gene in rows, sample in cols

ctrl <- cfg_prog$rna
ngs_assay <- "allmirna"
ctrl_wgcna <- cfg_prog$rna$wgcna$baseline$allmirna

# Specific to sample_clustering
col_fill <- "glu120"
```

```
dta <- besd_mae |>
  (\(mae) mae[ , , ngs_assay])() |>
  filter_for_depth(ngs_assay, ctrl$depth_threshold[[ngs_assay]]) |>
  filter_for_coldata("avisit", is = "Baseline") |>
  filter_for_coldata("diabcd", is = c("NGT", "T2D")) |>
  filter_for_low_expr(ngs_assay, ctrl$cpm_threshold[[ngs_assay]]) |>
  (\(mae) {
    mae[[ngs_assay]] <-
      DESeq2::DESeqDataSetFromMatrix(
        countData = mae[[ngs_assay]],
        colData = colData(x = mae),
        design = stats::formula(~ diabcd)
      ) |>
      DESeq2::varianceStabilizingTransformation(blind = FALSE) |>
      SummarizedExperiment::assay()
    mae
  })()
```

```
Warning: 'experiments' dropped; see 'metadata'
```

```
harmonizing input:
  removing 282 sampleMap rows not in names(experiments)
```

```
factor levels were dropped which had no samples
```

```
#' WGCNA Expression data.
#'
```

```
#' A matrix or data frame in which columns are genes and rows ar samples.
#'
dta_expr <- t(dta[[ngs_assay]])
assert_that(WGCNA::goodSamplesGenes(dta_expr, verbose = 0)$allOK)
```

```
dta_traits <- colData(dta) |> droplevels()
```

6.0.2.1 Sample clustering (OID 1)

```
spl_clust <- hclust(dist(dta_expr), method = "complete")

# Clustering Dendrogram of samples based on their Euclidean distance
theme_custom <- function(...) {
    theme_dendro() +
    theme(
      plot.margin = unit(c(0, 0, 0, 0), "null"),
      panel.spacing = unit(c(0, 0, 0, 0), "null")
    ) +
    theme(...)
}

library(ggdendro)
library(ggplot2)
dend <- spl_clust |> as.dendrogram()

dta_traits$x <- rownames(dta_traits)
dta_traits$x <- factor(dta_traits$x, levels = labels(dend))
dta_traits <- as.data.frame(dta_traits)

lim <- c(0, nobs(dend))
ddata <- ggdendro::dendro_data(dend, type = "rectangle")

p1 <- ggplot(ggdendro::segment(ddata)) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend)) +
  coord_flip(clip = "off") +
  scale_y_reverse(expand = expansion(0)) +
  scale_x_continuous(
```

```
    breaks = seq_len(nobs(dend)),
    labels = ggdendro::label(ddata)$label,
    lim = lim,
    position = "top"
  ) +
  theme_custom(axis.text.y = element_text())

p2 <- ggplot(
  dta_traits,
  aes(
    xmin = 0, xmax = 1,
    ymin = as.numeric(x) - 1/2, ymax = as.numeric(x) + 1/2,
    fill = diabcd
  )
) +
  geom_rect(col = "white") +
  scale_y_continuous(breaks = 1:34) +
  scale_x_continuous(expand = expansion(0)) +
  scale_fill_viridis_d(option = "C", direction = -1, begin = .2, end = .8) +
  coord_cartesian(ylim = lim) +
  theme_custom(legend.position = "top")

p4 <- ggplot(
  dta_traits,
  aes(
    xmin = 0, xmax = 1,
    ymin = as.numeric(x) - 1/2, ymax = as.numeric(x) + 1/2,
    fill = !!as.name(col_fill)
  )
) +
  geom_rect(col = "white") +
  scale_y_continuous(breaks = 1:34) +
  scale_x_continuous(expand = expansion(0)) +
  scale_fill_viridis_c() +
  coord_cartesian(ylim = lim) +
  theme_custom(legend.position = "top")

p <- egg::ggarrange(
  p1, p2, p4,
  nrow = 1,
  widths = c(2, 1, 1),
```

```
  padding = unit(0, "lines"),
  draw = FALSE
)
```

```
oid <- 1
title <- "miRNA / Baseline - Clustering dendrogram of samples based on their
  Euclidean distance and trait heatmap"
```

```
p <- clean_slate() |>
  add_title(c(
    paste0("Figure 11.",   oid),
    wrap_long_lines(title)
  )) |>
  add_figure(p, height = inches(5)) |>
  add_note(c(
    "Note: VST transformed expression matrix used to estimate distances
    between samples, _complete_ method was used for the hierarchical
    clustering."
  ))

oid <- stringr::str_pad(oid, 2, "left", "0")
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, ".pdf")),
  file_graph_alone =
    file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, "_af.pdf"))
)
```

```
[log] output saved as: ../tlg/graph/t2d_11_01.pdf
```

```
[log] output saved as: ../tlg/graph/t2d_11_01_af.pdf (annot. free)
```

```
show_slate(p)
```

6.0.2.2 Gene Clustering

6.0.2.2.1 Soft threshold determination (OID 2)

```r
library(WGCNA)
```

```
Loading required package: dynamicTreeCut
```

```
Loading required package: fastcluster
```

```

Attaching package: 'fastcluster'
```

```
The following object is masked from 'package:stats':

    hclust
```

```

Attaching package: 'WGCNA'
```

```
The following object is masked from 'package:IRanges':

    cor
```

```
The following object is masked from 'package:S4Vectors':

    cor
```

```
The following object is masked from 'package:stats':

    cor
```

```r
enableWGCNAThreads()

pwr <- ctrl_wgcna$adjacency$power
sft <- pickSoftThreshold(
  dta_expr,
  powerVector = c(c(1:10), seq(from = 12, to=20, by=2)),
  verbose = 0
)

library(ggplot2)
```

```
p1 <- ggplot(
  sft$fitIndices,
  aes(Power, - sign(slope) * SFT.R.sq, label = Power)
) +
  geom_hline(yintercept = .9, lty = 2) +
  geom_hline(yintercept = c(0, 1), lty = 1) +
  geom_line() +
  geom_label(
    data = sft$fitIndices |> subset(Power == pwr),
    vjust = 0, nudge_y = .025
  ) +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = Power, yend = 0)
  ) +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = -Inf, yend = - sign(slope) * SFT.R.sq)
  ) +
  geom_point(aes(shape = Power == pwr), show.legend = FALSE, fill = "white") +
  scale_shape_manual(values = c("TRUE" = 21, "FALSE" = 3)) +
  coord_cartesian(ylim = c(0, 1)) +
  xlab("Soft Threshold (power)") +
  ylab("Scale Free Topology Model Fit") +
  ggtitle("Scale independence") +
  theme_minimal() +
  theme(panel.grid.minor = element_blank())

p2 <- ggplot(
  sft$fitIndices,
  aes(Power, mean.k., label = Power, shape = Power == pwr)) +
  geom_hline(yintercept = .9) +
  geom_line() +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = Power, yend = 0)
  ) +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = -Inf, yend = mean.k.)
  ) +
```

```
  geom_label(
    data = sft$fitIndices |> subset(Power == pwr),
    vjust = 0, hjust = 0, nudge_y = 1
  ) +
  geom_point(fill = "white", show.legend = FALSE) +
  scale_shape_manual(values = c("TRUE" = 21, "FALSE" = 3)) +
  xlab("Soft Threshold (power)") +
  ylab("Mean connectivity") +
  ggtitle("Mean connectivity") +
  theme_minimal() +
  theme(panel.grid.minor = element_blank())

p <- egg::ggarrange(p1, p2, nrow = 1, draw = FALSE)
```

```
oid <- 2
title <- "Micro-RNA / Baseline - Soft threshold determination"
```

```
p <- clean_slate() |>
  add_title(c(
    paste0("Figure 11.", oid),
    title
  )) |>
  add_figure(p, width = inches(6), height = inches(3))

oid <- stringr::str_pad(oid, 2, "left", "0")
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, ".pdf")),
  file_graph_alone =
    file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, "_af.pdf"))
)
```

```
[log] output saved as: ../tlg/graph/t2d_11_02.pdf
```

```
[log] output saved as: ../tlg/graph/t2d_11_02_af.pdf (annot. free)
```

```
show_slate(p)
```

6.0.2.2.2 Network

```r
# gene dissimilarity
tom_dis <- dta_expr |>
  adjacency(
    type = ctrl_wgcna$adjacency$type,
    power = ctrl_wgcna$adjacency$power,
    corFnc = ctrl_wgcna$adjacency$cor_fn$value
  ) |>
  TOMsimilarity(TOMType = ctrl_wgcna$tom$type) |>
  (\(tom) 1 - tom)()

gene_hc <- tom_dis |>
  as.dist() |>
  hclust(method = "average")

# gene clustering
dyn_mod <- cutreeDynamic(
  gene_hc,
  cutHeight = ctrl_wgcna$cuttree$cut_height,
  minClusterSize = ctrl_wgcna$cuttree$min_cluster_size,
  method = "hybrid",
  distM = tom_dis,
  deepSplit = ctrl_wgcna$cuttree$deep_split,
  pamRespectsDendro = FALSE
)
dyn_col <- labels2colors(dyn_mod)

# me: module eigengenes (also used to merge dyn_col)
dyn_mge <- dta_expr |> WGCNA::mergeCloseModules(dyn_col)
meig <- dyn_mge$newMEs
```

```r
# Not exported yet
WGCNA::plotDendroAndColors(
  dendro = gene_hc,
  colors = dyn_mge$color,
  dendroLabels = FALSE,
  addGuide = TRUE,
  guideHang = 0.05,
  groupLabels = "",
  cex.rowText = 0.5,
  cex.colorLabels = 0.5,
  cex.dendroLabels = 0.9,
```

```
  marAll = c(1, 1, 1, 1),
  axes = FALSE,
  ylab = NULL,
  main = NULL
)
graphics::par(mfrow = c(1, 1))
```

6.0.2.3 Eigengenes exploration

6.0.2.3.1 OID 3 - Heatmap

```
theme_custom_dend <- function(gg, ...) {
  gg +
    scale_x_continuous(expand = c(0, 0.5)) +
    scale_y_continuous(expand = c(0.02, 0)) +
    theme(
      axis.text.x = element_blank(),
      axis.text.y = element_blank(),
      legend.background = element_rect(
        fill = "transparent", colour = "transparent"
      ),
      panel.background = element_rect(
        fill = "transparent", colour = "transparent"
      ),
      plot.background = element_rect(
        fill = "transparent", colour = "transparent"
      )
    ) +
    theme(...)
}

#' @describeIn wrap_wgcna improved correlation matrix.
#' @param mat_y,mat_x the matrix used in rows or columns.
#' @param draw (`logical`)
#' @param newpage (`logical`)
#' @import ggplot2
#' @export
#'
gg_eigen <- function(mat_y, mat_x, draw = FALSE, newpage = FALSE) {
```

```r
dta <- stats::cor(mat_y, mat_x, use = "pairwise.complete.obs")
dta <- reshape2::melt(
  dta, varnames = c("eig", "traits"), value.name = "cor"
)
dta$cor_pval <- WGCNA::corPvalueStudent(dta$cor, nSamples = nrow(mat_x))

dendro_y   <- stats::hclust(d = stats::as.dist(1 - stats::cor(mat_y)))
dendro_y   <- stats::as.dendrogram(dendro_y);
y_neworder <- labels(dendro_y)
dendro_y   <- ggdendro::ggdendrogram(data = dendro_y, rotate = TRUE) +
  theme(plot.margin = unit(c(0,0,0,-0.2), units="lines"))
dendro_y <- theme_custom_dend(gg = dendro_y)

dendro_x <- stats::as.dist(1 - stats::cor(mat_x))
dendro_x <- stats::hclust(d = stats::dist(x = dendro_x))
dendro_x <- stats::as.dendrogram(dendro_x)
x_neworder <- labels(dendro_x)
dendro_x <- ggdendro::ggdendrogram(data = dendro_x, rotate = FALSE) +
  theme(plot.margin = unit(c(0,0,-.8,0), units="lines"))
dendro_x <- theme_custom_dend(gg = dendro_x)

dta$traits <- factor(dta$traits, levels = x_neworder)
dta$eig <- factor(dta$eig, levels = y_neworder)

dta$cor_pval_2d <- format(round(dta$cor_pval, 2), nsmall = 2)

# Create heatmap plot
heatmap.plot <- ggplot(
  data = dta,
  mapping = aes_string(x = "traits", y = "eig")
) + geom_tile(data = dta, fill = "transparent", col = "gray") +
  geom_tile(
    data = dta[dta$cor_pval < 0.1, ],
    mapping = aes_string(fill = "cor")
  ) + geom_text(
    data = dta[dta$cor_pval < 0.1, ],
    mapping = aes_string(label = "cor_pval_2d"),
    col = "white", size = 2
  ) + scale_fill_viridis_c(
    breaks = seq(-1, 1, .2),
    name = "Corr.",
```

```
    guide = guide_colourbar(barheight = 10),
    option = "B",
    direction = -1,
    lim = c(-1, 1),
    begin = .1,
    end = .9
  )+ coord_cartesian(
    expand = FALSE
  ) + theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    panel.grid.major = element_blank(),
    legend.position = "left",
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.background = element_rect(
      fill = "transparent", colour = "transparent"
    ),
    panel.background = element_rect(
      fill = "transparent", colour = "transparent"
    ),
    plot.background = element_rect(
      fill = "transparent", colour = "transparent"
    ),
    legend.key.width = grid::unit(.3, "cm"),
  )

a1 <- ggplotGrob(heatmap.plot)
a2 <- ggplotGrob(dendro_y)
a <- cbind(a1, a2, size = "max")
b <- ggplot2::ggplotGrob(dendro_x)
b <- gtable::gtable_add_cols(
  b,
  widths = a1$widths[1:(length(a1$widths)-length(b$widths))],
  pos = 0
)
b <- gtable::gtable_add_cols(b, widths = a2$widths, pos = -1)

ab <- rbind(b, a, size = "last")
ab <- rbind(b, a, size = "last", z = c(1,10))
ab$heights[7] <- unit(3, "lines")
ab$widths[16] <- unit(3, "lines")
```

```
  if (draw) {
    if (newpage) grid::grid.newpage()
    grid::grid.draw(ab)
  }

  list(
    gg = ab,
    data = dta
  )
}

prep_col <- function(x) UseMethod("prep_col", x)
prep_col.numeric <- function(x) identity(x)
prep_col.factor <- function(x) as.numeric(x)
prep_col.character <- function(x) {
  x <- as.factor(x)
  prep_col(x)
}
```

```
res <- gg_eigen(
  dta_traits |>
    subset(select = c(
      diabcd, diab, age, trainn,
      chol, glu0, glu120, glu30, glu60, hba1c, hdl,
      homab, homair, insulin0, insulin120, insulin30, insulin60, ldl,
      matsuda, trig, bmi, bodyfatp, dcal, dcarbt, dfatt,
      dprot, fatacfr, fatmass, leanmass, smmass, vat, vo2maxe,
      vo2maxlbm, vo2maxml, weight
    )) |>
    lapply(prep_col)|>
    DataFrame(row.names = rownames(dta_traits)) |>
    as.matrix(),
  meig
)
```

```
Scale for 'x' is already present. Adding another scale for 'x', which will
replace the existing scale.
```

```
Scale for 'y' is already present. Adding another scale for 'y', which will
replace the existing scale.
```

```
Scale for 'x' is already present. Adding another scale for 'x', which will
replace the existing scale.
```

```
Scale for 'y' is already present. Adding another scale for 'y', which will
replace the existing scale.
```

```
oid <- 3
```

```
p <- clean_slate() |>
  add_figure(res$gg)

oid <- stringr::str_pad(oid, 2, "left", "0")
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, ".pdf")),
  file_graph_alone =
    file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, "_af.pdf"))
)
```

```
[log] output saved as: ../tlg/graph/t2d_11_03.pdf
```

```
[log] output saved as: ../tlg/graph/t2d_11_03_af.pdf (annot. free)
```

```
show_slate(p)
```

6.0.2.3.2 OID 4 - Excel file

```
oid <- 4
title <- "miRNA at baseline - WGCNA modules"
```

```
spl_mod <- rbind(
  dyn_mge$oldMEs |>
    as.data.frame() |>
    tibble::rownames_to_column("sample") |>
    tidyr::pivot_longer(
      cols = -"sample",
      names_to = "module"
    ) |>
```

```
    within(type <- "raw_me"),
  dyn_mge$newMEs |>
    as.data.frame() |>
    tibble::rownames_to_column("sample") |>
    tidyr::pivot_longer(
      cols = -"sample",
      names_to = "module"
    ) |>
    within(type <- "merged_me")
)

ft_mod <- data.frame(
  feature = colnames(dta_expr),
  raw_me = dyn_col,
  merged_me = dyn_mge$colors
)

oid <- stringr::str_pad(oid, 2, "left", "0")

export_xl(
  ft_mod = ft_mod,
  spl_mod = spl_mod,
  info = c(
    title = title,
    ft_mod = wrap_long_lines(
      "Associate genetic features with calculated module (a feature belongs to
      a module)"
    ),
    spl_mod = "Sample eigengen value (eigengen _expression profile_)",
    source = "UMB-BESD"
  ),
  dir = cfg_prog$paths$grh,
  basenm = paste0(params$basenm, "_", oid),
  rdata = TRUE
)
```

```
RData output (binary: ../tlg/graph/t2d_11_04.RData

Excel output: ../tlg/graph/t2d_11_04.xlsx
```

6.0.3 mRNA / Baseline

```
#' # https://horvath.genetics.ucla.edu/html/CoexpressionNetwork/Rpackages/WGCNA/Tutorials/
#' # WGCNA: gene in rows, sample in cols

ctrl <- cfg_prog$rna
ngs_assay <- "mrna"
ctrl_wgcna <- cfg_prog$rna$wgcna$baseline$mrna
```

```
dta <- besd_mae |>
  (\(mae) mae[ , , ngs_assay])() |>
  filter_for_depth(ngs_assay, ctrl$depth_threshold[[ngs_assay]]) |>
  filter_for_coldata("avisit", is = "Baseline") |>
  filter_for_coldata("diabcd", is = c("NGT", "T2D")) |>
  filter_for_low_expr(ngs_assay, ctrl$cpm_threshold[[ngs_assay]]) |>
  (\(mae) {
    mae[[ngs_assay]] <-
      DESeq2::DESeqDataSetFromMatrix(
        countData = mae[[ngs_assay]],
        colData = colData(x = mae),
        design = stats::formula(~ diabcd)
      ) |>
      DESeq2::varianceStabilizingTransformation(blind = FALSE) |>
      SummarizedExperiment::assay()
    mae
  })()
```

```
Warning: 'experiments' dropped; see 'metadata'
```

```
harmonizing input:
  removing 282 sampleMap rows not in names(experiments)
```

```
factor levels were dropped which had no samples
```

```
#' WGCNA Expression data.
#'
#' A matrix or data frame in which columns are genes and rows ar samples.
#'
dta_expr <- t(dta[[ngs_assay]])
```

```
assert_that(WGCNA::goodSamplesGenes(dta_expr, verbose = 0)$allOK)

dta_traits <- colData(dta) |> droplevels()
```

```
expr_desc <- "mRNA / Baseline - VST"
exportas <- "dta_11_01.RData"
```

```
attr(dta_expr, "desc") <- expr_desc
attr(dta_expr, "label") <- "Variance-Stabilised Transformed expression matrix"

knitr::kable(dta_expr[1:5, 1:3])
filenm <- file.path(cfg_prog$paths$dta, exportas)
save(dta_expr, file = filenm)
message("Export: ", filenm)
```

```
Export: ../data/dta_11_01.RData
```

```
rm(expr_desc, exportas)
```

6.0.3.1 Sample clustering (OID 5)

```
spl_clust <- hclust(dist(dta_expr), method = "complete")

# Clustering Dendrogram of samples based on their Euclidean distance
theme_custom <- function(...) {
    theme_dendro() +
    theme(
      plot.margin = unit(c(0, 0, 0, 0), "null"),
      panel.spacing = unit(c(0, 0, 0, 0), "null")
    ) +
    theme(...)
}

library(ggdendro)
library(ggplot2)
dend <- spl_clust |> as.dendrogram()

dta_traits$x <- rownames(dta_traits)
```

```r
dta_traits$x <- factor(dta_traits$x, levels = labels(dend))
dta_traits <- as.data.frame(dta_traits)

lim <- c(0, nobs(dend))
ddata <- ggdendro::dendro_data(dend, type = "rectangle")

p1 <- ggplot(ggdendro::segment(ddata)) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend)) +
  coord_flip(clip = "off") +
  scale_y_reverse(expand = expansion(0)) +
  scale_x_continuous(
    breaks = seq_len(nobs(dend)),
    labels = ggdendro::label(ddata)$label,
    lim = lim,
    position = "top"
  ) +
  theme_custom(axis.text.y = element_text())

p2 <- ggplot(
  dta_traits,
  aes(
    xmin = 0, xmax = 1,
    ymin = as.numeric(x) - 1/2, ymax = as.numeric(x) + 1/2,
    fill = diabcd
  )
) +
  geom_rect(col = "white") +
  scale_y_continuous(breaks = 1:34) +
  scale_x_continuous(expand = expansion(0)) +
  scale_fill_viridis_d(option = "C", direction = -1, begin = .2, end = .8) +
  coord_cartesian(ylim = lim) +
  theme_custom(legend.position = "top")

p4 <- ggplot(
  dta_traits,
  aes(
    xmin = 0, xmax = 1,
    ymin = as.numeric(x) - 1/2, ymax = as.numeric(x) + 1/2,
    fill = !!as.name(col_fill)
  )
) +
```

```
  geom_rect(col = "white") +
  scale_y_continuous(breaks = 1:34) +
  scale_x_continuous(expand = expansion(0)) +
  scale_fill_viridis_c() +
  coord_cartesian(ylim = lim) +
  theme_custom(legend.position = "top")

p <- egg::ggarrange(
  p1, p2, p4,
  nrow = 1,
  widths = c(2, 1, 1),
  padding = unit(0, "lines"),
  draw = FALSE
)
```

```
oid <- 5
title <- "mRNA / Baseline -Clustering dendrogram of samples based on their
  Euclidean distance and trait heatmap"
```

```
p <- clean_slate() |>
  add_title(c(
    paste0("Figure 11.",   oid),
    wrap_long_lines(title)
  )) |>
  add_figure(p, height = inches(5)) |>
  add_note(c(
    "Note: VST transformed expression matrix used to estimate distances
    between samples, _complete_ method was used for the hierarchical
    clustering."
  ))

oid <- stringr::str_pad(oid, 2, "left", "0")
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, ".pdf")),
  file_graph_alone =
    file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, "_af.pdf"))
)
```

```
[log] output saved as: ../tlg/graph/t2d_11_05.pdf
```

```
[log] output saved as: ../tlg/graph/t2d_11_05_af.pdf (annot. free)
```

```
show_slate(p)
```

6.0.3.2 Gene Clustering

6.0.3.2.1 Soft threshold determination (OID 6)

```
library(WGCNA)
enableWGCNAThreads()

pwr <- ctrl_wgcna$adjacency$power
sft <- pickSoftThreshold(
  dta_expr,
  powerVector = c(c(1:10), seq(from = 12, to=20, by=2)),
  verbose = 0
)

library(ggplot2)
p1 <- ggplot(
  sft$fitIndices,
  aes(Power, - sign(slope) * SFT.R.sq, label = Power)
) +
  geom_hline(yintercept = .9, lty = 2) +
  geom_hline(yintercept = c(0, 1), lty = 1) +
  geom_line() +
  geom_label(
    data = sft$fitIndices |> subset(Power == pwr),
    vjust = 0, nudge_y = .025
  ) +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = Power, yend = 0)
  ) +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = -Inf, yend = - sign(slope) * SFT.R.sq)
  ) +
  geom_point(aes(shape = Power == pwr), show.legend = FALSE, fill = "white") +
  scale_shape_manual(values = c("TRUE" = 21, "FALSE" = 3)) +
```

```
  coord_cartesian(ylim = c(0, 1)) +
  xlab("Soft Threshold (power)") +
  ylab("Scale Free Topology Model Fit") +
  ggtitle("Scale independence") +
  theme_minimal() +
  theme(panel.grid.minor = element_blank())

p2 <- ggplot(
  sft$fitIndices,
  aes(Power, mean.k., label = Power, shape = Power == pwr)) +
  geom_hline(yintercept = .9) +
  geom_line() +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = Power, yend = 0)
  ) +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = -Inf, yend = mean.k.)
  ) +
  geom_label(
    data = sft$fitIndices |> subset(Power == pwr),
    vjust = 0, hjust = 0, nudge_y = 1
  ) +
  geom_point(fill = "white", show.legend = FALSE) +
  scale_shape_manual(values = c("TRUE" = 21, "FALSE" = 3)) +
  xlab("Soft Threshold (power)") +
  ylab("Mean connectivity") +
  ggtitle("Mean connectivity") +
  theme_minimal() +
  theme(panel.grid.minor = element_blank())

p <- egg::ggarrange(p1, p2, nrow = 1, draw = FALSE)
```

```
oid <- 6
title <- "mRNA / Baseline - Soft threshold determination"
```

```
p <- clean_slate() |>
  add_title(c(
    paste0("Figure 11.", oid),
    title
```

```
  )) |>
  add_figure(p, width = inches(6), height = inches(3))

oid <- stringr::str_pad(oid, 2, "left", "0")
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, ".pdf")),
  file_graph_alone =
    file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, "_af.pdf"))
)
```

```
[log] output saved as: ../tlg/graph/t2d_11_06.pdf
```

```
[log] output saved as: ../tlg/graph/t2d_11_06_af.pdf (annot. free)
```

```
show_slate(p)
```

6.0.3.2.2 Network

```
# gene dissimilarity
tom_dis <- dta_expr |>
  adjacency(
    type = ctrl_wgcna$adjacency$type,
    power = ctrl_wgcna$adjacency$power,
    corFnc = ctrl_wgcna$adjacency$cor_fn$value
  ) |>
  TOMsimilarity(TOMType = ctrl_wgcna$tom$type) |>
  (\(tom) 1 - tom)()

gene_hc <- tom_dis |>
  as.dist() |>
  hclust(method = "average")

# gene clustering
dyn_mod <- cutreeDynamic(
  gene_hc,
  cutHeight = ctrl_wgcna$cuttree$cut_height,
  minClusterSize = ctrl_wgcna$cuttree$min_cluster_size,
  method = "hybrid",
```

```
  distM = tom_dis,
  deepSplit = ctrl_wgcna$cuttree$deep_split,
  pamRespectsDendro = FALSE
)
dyn_col <- labels2colors(dyn_mod)

# me: module eigengenes (also used to merge dyn_col)
dyn_mge <- dta_expr |> WGCNA::mergeCloseModules(dyn_col)
meig <- dyn_mge$newMEs
```

```
# Not exported yet
WGCNA::plotDendroAndColors(
  dendro = gene_hc,
  colors = dyn_mge$color,
  dendroLabels = FALSE,
  addGuide = TRUE,
  guideHang = 0.05,
  groupLabels = "",
  cex.rowText = 0.5,
  cex.colorLabels = 0.5,
  cex.dendroLabels = 0.9,
  marAll = c(1, 1, 1, 1),
  axes = FALSE,
  ylab = NULL,
  main = NULL
)
graphics::par(mfrow = c(1, 1))
```

6.0.3.3 Eigengenes exploration

6.0.3.3.1 OID 7 - Heatmap

```
res <- gg_eigen(
  dta_traits |>
    subset(select = c(
      diabcd, diab, age, trainn,
      chol, glu0, glu120, glu30, glu60, hba1c, hdl,
      homab, homair, insulin0, insulin120, insulin30, insulin60, ldl,
      matsuda, trig, bmi, bodyfatp, dcal, dcarbt, dfatt,
      dprot, fatacfr, fatmass, leanmass, smmass, vat, vo2maxe,
```

```
      vo2maxlbm, vo2maxml, weight
    )) |>
    lapply(prep_col)|>
    DataFrame(row.names = rownames(dta_traits)) |>
    as.matrix(),
  meig
)
```

```
Scale for 'x' is already present. Adding another scale for 'x', which will
replace the existing scale.
```

```
Scale for 'y' is already present. Adding another scale for 'y', which will
replace the existing scale.
```

```
Scale for 'x' is already present. Adding another scale for 'x', which will
replace the existing scale.
```

```
Scale for 'y' is already present. Adding another scale for 'y', which will
replace the existing scale.
```

```
oid <- 7
```

```
p <- clean_slate() |>
  add_figure(res$gg)

oid <- stringr::str_pad(oid, 2, "left", "0")
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, ".pdf")),
  file_graph_alone =
    file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, "_af.pdf"))
)
```

```
[log] output saved as: ../tlg/graph/t2d_11_07.pdf
```

```
[log] output saved as: ../tlg/graph/t2d_11_07_af.pdf (annot. free)
```

```
show_slate(p)
```

6.0.3.3.2 OID 8 - Excel file

```
oid <- 8
title <- "mRNA at baseline - WGCNA modules"
```

```
spl_mod <- rbind(
  dyn_mge$oldMEs |>
    as.data.frame() |>
    tibble::rownames_to_column("sample") |>
    tidyr::pivot_longer(
      cols = -"sample",
      names_to = "module"
    ) |>
    within(type <- "raw_me"),
  dyn_mge$newMEs |>
    as.data.frame() |>
    tibble::rownames_to_column("sample") |>
    tidyr::pivot_longer(
      cols = -"sample",
      names_to = "module"
    ) |>
    within(type <- "merged_me")
)

ft_mod <- data.frame(
  feature = colnames(dta_expr),
  raw_me = dyn_col,
  merged_me = dyn_mge$colors
)

oid <- stringr::str_pad(oid, 2, "left", "0")

export_xl(
  ft_mod = ft_mod,
  spl_mod = spl_mod,
  info = c(
    title = title,
    ft_mod = wrap_long_lines(
      "Associate genetic features with calculated module (a feature belongs to
      a module)"
    ),
```

```
    spl_mod = "Sample eigengen value (eigengen _expression profile_)",
    source = "UMB-BESD"
  ),
  dir = cfg_prog$paths$grh,
  basenm = paste0(params$basenm, "_", oid),
  rdata = TRUE
)
```

```
RData output (binary: ../tlg/graph/t2d_11_08.RData

Excel output: ../tlg/graph/t2d_11_08.xlsx
```

6.0.4 miRNA / CHG

```
#' # https://horvath.genetics.ucla.edu/html/CoexpressionNetwork/Rpackages/WGCNA/Tutorials/
#' # WGCNA: gene in rows, sample in cols

ctrl <- cfg_prog$rna
ngs_assay <- "allmirna"
ctrl_wgcna <- cfg_prog$rna$wgcna$chg$allmirna # TODO

# Specific to sample_clustering
col_fill <- "glu120_chg"
```

```
filter_for_paired_visits <- function(mae) {
  assert_that(all(table(colData(mae)$subjid) <= 2))

  subj <- table(colData(mae)$subjid)
  subj <- names(subj[subj == 2])
  mae <- mae[, colData(mae)$subjid %in% subj, ]

  chk_df <- as.data.frame(colData(mae))
  check <- split(chk_df, f = chk_df$avisit)

  # Verify the selection:
  assert_that(all(
    sapply(check, subj = subj,\(x, subj) all(subj %in% x$subjid))
  ))
  mae
```

```
}

gc()
dta <- besd_mae |>
  (\(mae) mae[ , , ngs_assay])() |>
  filter_for_depth(ngs_assay, ctrl$depth_threshold[[ngs_assay]]) |>
  filter_for_paired_visits() |>
  filter_for_low_expr(ngs_assay, ctrl$cpm_threshold[[ngs_assay]]) |>
  (\(mae) {
    # VST transformation
    colData(mae) <- droplevels(colData(mae))
    colData(mae)$avisit_ <- factor(
      colData(mae)$avisit,
      levels = levels(colData(mae)$avisit),
      labels = sub(" ", "_", levels(colData(mae)$avisit))
    )
    coldata <- colData(mae)[c("diabcd", "subjid", "avisit_")]

    mae[[ngs_assay]] <- DESeq2::DESeqDataSetFromMatrix(
      countData = mae[[ngs_assay]],
      colData = colData(x = mae),
      design = stats::formula(~ subjid + avisit_)
    ) |>
      DESeq2::varianceStabilizingTransformation(blind = FALSE) |>
      SummarizedExperiment::assay()

    mae
  })() |>
  filter_for_coldata("diabcd", is = c("NGT", "T2D")) |>
  (\(mae) {
    # Get traits: BL, M3, CHG:
    bl <- mae[, colData(mae)$avisit == "Baseline", ]
    m3 <- mae[, colData(mae)$avisit == "Month 3", ]
    assert_that(all(colData(bl)$subjid == colData(m3)$subjid))
    mat <- m3[[ngs_assay]] - bl[[ngs_assay]]
    nm <- setNames(colData(mae)$subjid, nm = rownames(colData(mae)))
    colnames(mat) <- nm[colnames(mat)]

    get_vars_when <- function(mae,
                              fun = is.numeric,
                              col_names = TRUE) {
```

```
      df <- colData(mae)
      assert_that(all(table(df$subjid) <= 1))
      rownames(df) <- df$subjid
      df <- df[, sapply(df, fun)]
      if (col_names)
        colnames(df) <- paste0(tolower(colnames(df)), "_", substitute(mae))
      df
    }
    m3_cd <- get_vars_when(m3, is.numeric) # m3_col data
    bl_cd <- get_vars_when(bl, is.numeric)
    # bl qualitative (diabcd and other)
    bl_ql <- get_vars_when(bl, Negate(is.numeric), col_names = FALSE)
    traits <- Map(a = m3_cd, b = bl_cd, f = \(a, b) a - b) |>
      S4Vectors::DataFrame() |>
      (\(df, ...) {
        args <- list(...)
        colnames(df) <- gsub("_m3", "_chg", colnames(df))
        df <- do.call(cbind, c(args, list(df)))
        # Constant columns are not interesting for gene variation association
        df[sapply(df, sd) > 0]
      })(bl_cd, m3_cd) |>
      cbind(bl_ql)

    MultiAssayExperiment(
      experiments = ExperimentList(vst = mat),
      colData = traits
    )
  })()
```

```
Warning: 'experiments' dropped; see 'metadata'
```

```
harmonizing input:
  removing 282 sampleMap rows not in names(experiments)
```

```
#' WGCNA Expression data.
#'
#' A matrix or data frame in which columns are genes and rows ar samples.
#'
dta_expr <- t(dta[["vst"]])
assert_that(WGCNA::goodSamplesGenes(dta_expr, verbose = 0)$allOK)
```

```
dta_traits <- colData(dta) |> droplevels()
```

6.0.4.1 Sample Clustering (OID 9)

```
spl_clust <- hclust(dist(dta_expr), method = "complete")

# Clustering Dendrogram of samples based on their Euclidean distance
theme_custom <- function(...) {
    theme_dendro() +
    theme(
      plot.margin = unit(c(0, 0, 0, 0), "null"),
      panel.spacing = unit(c(0, 0, 0, 0), "null")
    ) +
    theme(...)
}

library(ggdendro)
library(ggplot2)
dend <- spl_clust |> as.dendrogram()

dta_traits$x <- rownames(dta_traits)
dta_traits$x <- factor(dta_traits$x, levels = labels(dend))
dta_traits <- as.data.frame(dta_traits)

lim <- c(0, nobs(dend))
ddata <- ggdendro::dendro_data(dend, type = "rectangle")

p1 <- ggplot(ggdendro::segment(ddata)) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend)) +
  coord_flip(clip = "off") +
  scale_y_reverse(expand = expansion(0)) +
  scale_x_continuous(
    breaks = seq_len(nobs(dend)),
    labels = ggdendro::label(ddata)$label,
    lim = lim,
    position = "top"
  ) +
  theme_custom(axis.text.y = element_text())

p2 <- ggplot(
```

```
  dta_traits,
  aes(
    xmin = 0, xmax = 1,
    ymin = as.numeric(x) - 1/2, ymax = as.numeric(x) + 1/2,
    fill = diabcd
  )
) +
  geom_rect(col = "white") +
  scale_y_continuous(breaks = 1:34) +
  scale_x_continuous(expand = expansion(0)) +
  scale_fill_viridis_d(option = "C", direction = -1, begin = .2, end = .8) +
  coord_cartesian(ylim = lim) +
  theme_custom(legend.position = "top")

p4 <- ggplot(
  dta_traits,
  aes(
    xmin = 0, xmax = 1,
    ymin = as.numeric(x) - 1/2, ymax = as.numeric(x) + 1/2,
    fill = !!as.name(col_fill)
  )
) +
  geom_rect(col = "white") +
  scale_y_continuous(breaks = 1:34) +
  scale_x_continuous(expand = expansion(0)) +
  scale_fill_viridis_c() +
  coord_cartesian(ylim = lim) +
  theme_custom(legend.position = "top")

p <- egg::ggarrange(
  p1, p2, p4,
  nrow = 1,
  widths = c(2, 1, 1),
  padding = unit(0, "lines"),
  draw = FALSE
)
```

```
oid <- 9
title <- "miRNA / Change from Baseline - Clustering dendrogram of samples based
  on their Euclidean distance and trait heatmap"
```

```
p <- clean_slate() |>
  add_title(c(
    paste0("Figure 11.",   oid),
    wrap_long_lines(title)
  )) |>
  add_figure(p, height = inches(5)) |>
  add_note(c(
    "Note: VST transformed expression matrix used to estimate distances
    between samples, _complete_ method was used for the hierarchical
    clustering."
  ))

oid <- stringr::str_pad(oid, 2, "left", "0")
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, ".pdf")),
  file_graph_alone =
    file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, "_af.pdf"))
)
```

```
[log] output saved as: ../tlg/graph/t2d_11_09.pdf
```

```
[log] output saved as: ../tlg/graph/t2d_11_09_af.pdf (annot. free)
```

```
show_slate(p)
```

6.0.4.2 Gene Clustering

6.0.4.2.1 Soft threshold determination (OID 10)

```
library(WGCNA)
enableWGCNAThreads()

pwr <- ctrl_wgcna$adjacency$power
sft <- pickSoftThreshold(
  dta_expr,
  powerVector = c(c(1:10), seq(from = 12, to=20, by=2)),
  verbose = 0
)
```

```r
library(ggplot2)
p1 <- ggplot(
  sft$fitIndices,
  aes(Power, - sign(slope) * SFT.R.sq, label = Power)
) +
  geom_hline(yintercept = .9, lty = 2) +
  geom_hline(yintercept = c(0, 1), lty = 1) +
  geom_line() +
  geom_label(
    data = sft$fitIndices |> subset(Power == pwr),
    vjust = 0, nudge_y = .025
  ) +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = Power, yend = 0)
  ) +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = -Inf, yend = - sign(slope) * SFT.R.sq)
  ) +
  geom_point(aes(shape = Power == pwr), show.legend = FALSE, fill = "white") +
  scale_shape_manual(values = c("TRUE" = 21, "FALSE" = 3)) +
  coord_cartesian(ylim = c(0, 1)) +
  xlab("Soft Threshold (power)") +
  ylab("Scale Free Topology Model Fit") +
  ggtitle("Scale independence") +
  theme_minimal() +
  theme(panel.grid.minor = element_blank())

p2 <- ggplot(
  sft$fitIndices,
  aes(Power, mean.k., label = Power, shape = Power == pwr)) +
  geom_hline(yintercept = .9) +
  geom_line() +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = Power, yend = 0)
  ) +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = -Inf, yend = mean.k.)
```

```
  ) +
  geom_label(
    data = sft$fitIndices |> subset(Power == pwr),
    vjust = 0, hjust = 0, nudge_y = 1
  ) +
  geom_point(fill = "white", show.legend = FALSE) +
  scale_shape_manual(values = c("TRUE" = 21, "FALSE" = 3)) +
  xlab("Soft Threshold (power)") +
  ylab("Mean connectivity") +
  ggtitle("Mean connectivity") +
  theme_minimal() +
  theme(panel.grid.minor = element_blank())

p <- egg::ggarrange(p1, p2, nrow = 1, draw = FALSE)
```

```
oid <- 10
title <- "Micro-RNA / Change from Baseline - Soft threshold determination"
```

```
p <- clean_slate() |>
  add_title(c(
    paste0("Figure 11.", oid),
    title
  )) |>
  add_figure(p, width = inches(6), height = inches(3))

oid <- stringr::str_pad(oid, 2, "left", "0")
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, ".pdf")),
  file_graph_alone =
    file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, "_af.pdf"))
)
```

```
[log] output saved as: ../tlg/graph/t2d_11_10.pdf
```

```
[log] output saved as: ../tlg/graph/t2d_11_10_af.pdf (annot. free)
```

```
show_slate(p)
```

6.0.4.2.2 Network

```
# gene dissimilarity
tom_dis <- dta_expr |>
  adjacency(
    type = ctrl_wgcna$adjacency$type,
    power = ctrl_wgcna$adjacency$power,
    corFnc = ctrl_wgcna$adjacency$cor_fn$value
  ) |>
  TOMsimilarity(TOMType = ctrl_wgcna$tom$type) |>
  (\(tom) 1 - tom)()

gene_hc <- tom_dis |>
  as.dist() |>
  hclust(method = "average")

# gene clustering
dyn_mod <- cutreeDynamic(
  gene_hc,
  cutHeight = ctrl_wgcna$cuttree$cut_height,
  minClusterSize = ctrl_wgcna$cuttree$min_cluster_size,
  method = "hybrid",
  distM = tom_dis,
  deepSplit = ctrl_wgcna$cuttree$deep_split,
  pamRespectsDendro = FALSE
)
dyn_col <- labels2colors(dyn_mod)

# me: module eigengenes (also used to merge dyn_col)
dyn_mge <- dta_expr |> WGCNA::mergeCloseModules(dyn_col)
meig <- dyn_mge$newMEs
```

```
# Not exported yet
WGCNA::plotDendroAndColors(
  dendro = gene_hc,
  colors = dyn_mge$color,
  dendroLabels = FALSE,
  addGuide = TRUE,
  guideHang = 0.05,
  groupLabels = "",
  cex.rowText = 0.5,
```

```
  cex.colorLabels = 0.5,
  cex.dendroLabels = 0.9,
  marAll = c(1, 1, 1, 1),
  axes = FALSE,
  ylab = NULL,
  main = NULL
)
graphics::par(mfrow = c(1, 1))
```

6.0.4.3 Eigengenes exploration

6.0.4.3.1 OID 11 - Heatmap

```
res <- gg_eigen(
  dta_traits |>
    (\(df) df[sapply(
       df,
       function(x) if (is.numeric(x)) sd(x) > 0 else length(unique(x)) > 1
     )]) () |>
    lapply(prep_col)|>
    DataFrame(row.names = rownames(dta_traits)) |>
    as.matrix(),
  meig
)
```

```
Scale for 'x' is already present. Adding another scale for 'x', which will
replace the existing scale.
```

```
Scale for 'y' is already present. Adding another scale for 'y', which will
replace the existing scale.
```

```
Scale for 'x' is already present. Adding another scale for 'x', which will
replace the existing scale.
```

```
Scale for 'y' is already present. Adding another scale for 'y', which will
replace the existing scale.
```

```
oid <- 11
```

```
p <- clean_slate() |>
  add_figure(res$gg)

oid <- stringr::str_pad(oid, 2, "left", "0")
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, ".pdf")),
  file_graph_alone =
    file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, "_af.pdf"))
)
```

```
[log] output saved as: ../tlg/graph/t2d_11_11.pdf
```

```
[log] output saved as: ../tlg/graph/t2d_11_11_af.pdf (annot. free)
```

```
show_slate(p)
```

6.0.4.3.2 OID 12 - Excel file

```
oid <- 12
title <- "miRNA Change from Baseline - WGCNA modules"
```

```
spl_mod <- rbind(
  dyn_mge$oldMEs |>
    as.data.frame() |>
    tibble::rownames_to_column("sample") |>
    tidyr::pivot_longer(
      cols = -"sample",
      names_to = "module"
    ) |>
    within(type <- "raw_me"),
  dyn_mge$newMEs |>
    as.data.frame() |>
    tibble::rownames_to_column("sample") |>
    tidyr::pivot_longer(
      cols = -"sample",
      names_to = "module"
    ) |>
    within(type <- "merged_me")
```

```
)

ft_mod <- data.frame(
  feature = colnames(dta_expr),
  raw_me = dyn_col,
  merged_me = dyn_mge$colors
)

oid <- stringr::str_pad(oid, 2, "left", "0")

export_xl(
  ft_mod = ft_mod,
  spl_mod = spl_mod,
  info = c(
    title = title,
    ft_mod = wrap_long_lines(
      "Associate genetic features with calculated module (a feature belongs to
      a module)"
    ),
    spl_mod = "Sample eigengen value (eigengen _expression profile_)",
    source = "UMB-BESD"
  ),
  dir = cfg_prog$paths$grh,
  basenm = paste0(params$basenm, "_", oid),
  rdata = TRUE
)
```

```
RData output (binary: ../tlg/graph/t2d_11_12.RData

Excel output: ../tlg/graph/t2d_11_12.xlsx
```

6.0.5 mRNA / CHG

```
#' # https://horvath.genetics.ucla.edu/html/CoexpressionNetwork/Rpackages/WGCNA/Tutorials/
#' # WGCNA: gene in rows, sample in cols

ctrl <- cfg_prog$rna
ngs_assay <- "mrna"
ctrl_wgcna <- cfg_prog$rna$wgcna$chg$mrna
```

```
# Specific to sample_clustering
col_fill <- "glu120_chg"
```

```
filter_for_paired_visits <- function(mae) {
  assert_that(all(table(colData(mae)$subjid) <= 2))

  subj <- table(colData(mae)$subjid)
  subj <- names(subj[subj == 2])
  mae <- mae[, colData(mae)$subjid %in% subj, ]

  chk_df <- as.data.frame(colData(mae))
  check <- split(chk_df, f = chk_df$avisit)

  # Verify the selection:
  assert_that(all(
    sapply(check, subj = subj,\(x, subj) all(subj %in% x$subjid))
  ))
  mae
}

gc()
dta <- besd_mae |>
  (\(mae) mae[ , , ngs_assay])() |>
  filter_for_depth(ngs_assay, ctrl$depth_threshold[[ngs_assay]]) |>
  filter_for_paired_visits() |>
  filter_for_low_expr(ngs_assay, ctrl$cpm_threshold[[ngs_assay]]) |>
  (\(mae) {
    # VST transformation
    colData(mae) <- droplevels(colData(mae))
    colData(mae)$avisit_ <- factor(
      colData(mae)$avisit,
      levels = levels(colData(mae)$avisit),
      labels = sub(" ", "_", levels(colData(mae)$avisit))
    )
    coldata <- colData(mae)[c("diabcd", "subjid", "avisit_")]

    mae[[ngs_assay]] <- DESeq2::DESeqDataSetFromMatrix(
      countData = mae[[ngs_assay]],
      colData = colData(x = mae),
      design = stats::formula(~ subjid + avisit_)
    ) |>
```

```
    DESeq2::varianceStabilizingTransformation(blind = FALSE) |>
    SummarizedExperiment::assay()

  mae
})() |>
filter_for_coldata("diabcd", is = c("NGT", "T2D")) |>
(\(mae) {
  # Get traits: BL, M3, CHG:
  bl <- mae[, colData(mae)$avisit == "Baseline", ]
  m3 <- mae[, colData(mae)$avisit == "Month 3", ]
  assert_that(all(colData(bl)$subjid == colData(m3)$subjid))
  mat <- m3[[ngs_assay]] - bl[[ngs_assay]]
  nm <- setNames(colData(mae)$subjid, nm = rownames(colData(mae)))
  colnames(mat) <- nm[colnames(mat)]

  get_vars_when <- function(mae,
                            fun = is.numeric,
                            col_names = TRUE) {
    df <- colData(mae)
    assert_that(all(table(df$subjid) <= 1))
    rownames(df) <- df$subjid
    df <- df[, sapply(df, fun)]
    if (col_names)
      colnames(df) <- paste0(tolower(colnames(df)), "_", substitute(mae))
    df
  }
  m3_cd <- get_vars_when(m3, is.numeric) # m3_col data
  bl_cd <- get_vars_when(bl, is.numeric)
  # bl qualitative (diabcd and other)
  bl_ql <- get_vars_when(bl, Negate(is.numeric), col_names = FALSE)
  traits <- Map(a = m3_cd, b = bl_cd, f = \(a, b) a - b) |>
    S4Vectors::DataFrame() |>
    (\(df, ...) {
      args <- list(...)
      colnames(df) <- gsub("_m3", "_chg", colnames(df))
      df <- do.call(cbind, c(args, list(df)))
      # Constant columns are not interesting for gene variation association
      df[sapply(df, sd) > 0]
    })(bl_cd, m3_cd) |>
    cbind(bl_ql)
```

```
    MultiAssayExperiment(
      experiments = ExperimentList(vst = mat),
      colData = traits
    )
  })()
```

```
Warning: 'experiments' dropped; see 'metadata'
```

```
harmonizing input:
  removing 282 sampleMap rows not in names(experiments)
```

```
#' WGCNA Expression data.
#'
#' A matrix or data frame in which columns are genes and rows ar samples.
#'
dta_expr <- t(dta[["vst"]])
assert_that(WGCNA::goodSamplesGenes(dta_expr, verbose = 0)$allOK)
dta_traits <- colData(dta) |> droplevels()
```

6.0.5.1 Sample Clustering (OID 13)

```
spl_clust <- hclust(dist(dta_expr), method = "complete")

# Clustering Dendrogram of samples based on their Euclidean distance
theme_custom <- function(...) {
    theme_dendro() +
    theme(
      plot.margin = unit(c(0, 0, 0, 0), "null"),
      panel.spacing = unit(c(0, 0, 0, 0), "null")
    ) +
    theme(...)
}

library(ggdendro)
library(ggplot2)
dend <- spl_clust |> as.dendrogram()

dta_traits$x <- rownames(dta_traits)
dta_traits$x <- factor(dta_traits$x, levels = labels(dend))
```

```r
dta_traits <- as.data.frame(dta_traits)

lim <- c(0, nobs(dend))
ddata <- ggdendro::dendro_data(dend, type = "rectangle")

p1 <- ggplot(ggdendro::segment(ddata)) +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend)) +
  coord_flip(clip = "off") +
  scale_y_reverse(expand = expansion(0)) +
  scale_x_continuous(
    breaks = seq_len(nobs(dend)),
    labels = ggdendro::label(ddata)$label,
    lim = lim,
    position = "top"
  ) +
  theme_custom(axis.text.y = element_text())

p2 <- ggplot(
  dta_traits,
  aes(
    xmin = 0, xmax = 1,
    ymin = as.numeric(x) - 1/2, ymax = as.numeric(x) + 1/2,
    fill = diabcd
  )
) +
  geom_rect(col = "white") +
  scale_y_continuous(breaks = 1:34) +
  scale_x_continuous(expand = expansion(0)) +
  scale_fill_viridis_d(option = "C", direction = -1, begin = .2, end = .8) +
  coord_cartesian(ylim = lim) +
  theme_custom(legend.position = "top")

p4 <- ggplot(
  dta_traits,
  aes(
    xmin = 0, xmax = 1,
    ymin = as.numeric(x) - 1/2, ymax = as.numeric(x) + 1/2,
    fill = !!as.name(col_fill)
  )
) +
  geom_rect(col = "white") +
```

```
  scale_y_continuous(breaks = 1:34) +
  scale_x_continuous(expand = expansion(0)) +
  scale_fill_viridis_c() +
  coord_cartesian(ylim = lim) +
  theme_custom(legend.position = "top")

p <- egg::ggarrange(
  p1, p2, p4,
  nrow = 1,
  widths = c(2, 1, 1),
  padding = unit(0, "lines"),
  draw = FALSE
)
```

```
oid <- 13
title <- "mRNA / Change from Baseline - Clustering dendrogram of samples based
  on their Euclidean distance and trait heatmap"
```

```
p <- clean_slate() |>
  add_title(c(
    paste0("Figure 11.",   oid),
    wrap_long_lines(title)
  )) |>
  add_figure(p, height = inches(5)) |>
  add_note(c(
    "Note: VST transformed expression matrix used to estimate distances
    between samples, _complete_ method was used for the hierarchical
    clustering."
  ))

oid <- stringr::str_pad(oid, 2, "left", "0")
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, ".pdf")),
  file_graph_alone =
    file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, "_af.pdf"))
)
```

```
[log] output saved as: ../tlg/graph/t2d_11_13.pdf

[log] output saved as: ../tlg/graph/t2d_11_13_af.pdf (annot. free)
```

```
show_slate(p)
```

6.0.5.2 Gene Clustering

6.0.5.2.1 Soft threshold determination (OID 14)

```
library(WGCNA)
enableWGCNAThreads()

pwr <- ctrl_wgcna$adjacency$power
sft <- pickSoftThreshold(
  dta_expr,
  powerVector = c(c(1:10), seq(from = 12, to=20, by=2)),
  verbose = 0
)

library(ggplot2)
p1 <- ggplot(
  sft$fitIndices,
  aes(Power, - sign(slope) * SFT.R.sq, label = Power)
) +
  geom_hline(yintercept = .9, lty = 2) +
  geom_hline(yintercept = c(0, 1), lty = 1) +
  geom_line() +
  geom_label(
    data = sft$fitIndices |> subset(Power == pwr),
    vjust = 0, nudge_y = .025
  ) +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = Power, yend = 0)
  ) +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = -Inf, yend = - sign(slope) * SFT.R.sq)
  ) +
  geom_point(aes(shape = Power == pwr), show.legend = FALSE, fill = "white") +
  scale_shape_manual(values = c("TRUE" = 21, "FALSE" = 3)) +
  coord_cartesian(ylim = c(0, 1)) +
  xlab("Soft Threshold (power)") +
```

```
  ylab("Scale Free Topology Model Fit") +
  ggtitle("Scale independence") +
  theme_minimal() +
  theme(panel.grid.minor = element_blank())

p2 <- ggplot(
  sft$fitIndices,
  aes(Power, mean.k., label = Power, shape = Power == pwr)) +
  geom_hline(yintercept = .9) +
  geom_line() +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = Power, yend = 0)
  ) +
  geom_segment(
    data = sft$fitIndices |> subset(Power == pwr),
    aes(xend = -Inf, yend = mean.k.)
  ) +
  geom_label(
    data = sft$fitIndices |> subset(Power == pwr),
    vjust = 0, hjust = 0, nudge_y = 1
  ) +
  geom_point(fill = "white", show.legend = FALSE) +
  scale_shape_manual(values = c("TRUE" = 21, "FALSE" = 3)) +
  xlab("Soft Threshold (power)") +
  ylab("Mean connectivity") +
  ggtitle("Mean connectivity") +
  theme_minimal() +
  theme(panel.grid.minor = element_blank())

p <- egg::ggarrange(p1, p2, nrow = 1, draw = FALSE)
```

```
oid <- 14
title <- "mRNA / Change from Baseline - Soft threshold determination"
```

```
p <- clean_slate() |>
  add_title(c(
    paste0("Figure 11.", oid),
    title
  )) |>
```

```
  add_figure(p, width = inches(6), height = inches(3))

oid <- stringr::str_pad(oid, 2, "left", "0")
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, ".pdf")),
  file_graph_alone =
    file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, "_af.pdf"))
)
```

```
[log] output saved as: ../tlg/graph/t2d_11_14.pdf
```

```
[log] output saved as: ../tlg/graph/t2d_11_14_af.pdf (annot. free)
```

```
show_slate(p)
```

6.0.5.2.2 Network

```
# gene dissimilarity
tom_dis <- dta_expr |>
  adjacency(
    type = ctrl_wgcna$adjacency$type,
    power = ctrl_wgcna$adjacency$power,
    corFnc = ctrl_wgcna$adjacency$cor_fn$value
  ) |>
  TOMsimilarity(TOMType = ctrl_wgcna$tom$type) |>
  (\(tom) 1 - tom)()

gene_hc <- tom_dis |>
  as.dist() |>
  hclust(method = "average")

# gene clustering
dyn_mod <- cutreeDynamic(
  gene_hc,
  cutHeight = ctrl_wgcna$cuttree$cut_height,
  minClusterSize = ctrl_wgcna$cuttree$min_cluster_size,
  method = "hybrid",
  distM = tom_dis,
```

```
  deepSplit = ctrl_wgcna$cuttree$deep_split,
  pamRespectsDendro = FALSE
)
dyn_col <- labels2colors(dyn_mod)

# me: module eigengenes (also used to merge dyn_col)
dyn_mge <- dta_expr |> WGCNA::mergeCloseModules(dyn_col)
meig <- dyn_mge$newMEs
```

```
# Not exported yet
WGCNA::plotDendroAndColors(
  dendro = gene_hc,
  colors = dyn_mge$color,
  dendroLabels = FALSE,
  addGuide = TRUE,
  guideHang = 0.05,
  groupLabels = "",
  cex.rowText = 0.5,
  cex.colorLabels = 0.5,
  cex.dendroLabels = 0.9,
  marAll = c(1, 1, 1, 1),
  axes = FALSE,
  ylab = NULL,
  main = NULL
)
graphics::par(mfrow = c(1, 1))
```

6.0.5.3 Eigengenes exploration

6.0.5.3.1 OID 15 - Heatmap

```
res <- gg_eigen(
  dta_traits |>
    (\(df) df[sapply(
       df,
       function(x) if (is.numeric(x)) sd(x) > 0 else length(unique(x)) > 1
     )]) () |>
    lapply(prep_col)|>
    DataFrame(row.names = rownames(dta_traits)) |>
    as.matrix(),
```

```
  meig
)
```

```
Scale for 'x' is already present. Adding another scale for 'x', which will
replace the existing scale.
```

```
Scale for 'y' is already present. Adding another scale for 'y', which will
replace the existing scale.
```

```
Scale for 'x' is already present. Adding another scale for 'x', which will
replace the existing scale.
```

```
Scale for 'y' is already present. Adding another scale for 'y', which will
replace the existing scale.
```

```
oid <- 15
```

```
p <- clean_slate() |>
  add_figure(res$gg)

oid <- stringr::str_pad(oid, 2, "left", "0")
export_as(
  p,
  file = file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, ".pdf")),
  file_graph_alone =
    file.path(cfg_prog$paths$grh, paste0(params$basenm, "_", oid, "_af.pdf"))
)
```

```
[log] output saved as: ../tlg/graph/t2d_11_15.pdf
```

```
[log] output saved as: ../tlg/graph/t2d_11_15_af.pdf (annot. free)
```

```
show_slate(p)
```

6.0.5.3.2 OID 16 - Excel file

```r
oid <- 16
title <- "mRNA Change from Baseline - WGCNA modules"
```

```r
spl_mod <- rbind(
  dyn_mge$oldMEs |>
    as.data.frame() |>
    tibble::rownames_to_column("sample") |>
    tidyr::pivot_longer(
      cols = -"sample",
      names_to = "module"
    ) |>
    within(type <- "raw_me"),
  dyn_mge$newMEs |>
    as.data.frame() |>
    tibble::rownames_to_column("sample") |>
    tidyr::pivot_longer(
      cols = -"sample",
      names_to = "module"
    ) |>
    within(type <- "merged_me")
)

ft_mod <- data.frame(
  feature = colnames(dta_expr),
  raw_me = dyn_col,
  merged_me = dyn_mge$colors
)

oid <- stringr::str_pad(oid, 2, "left", "0")

export_xl(
  ft_mod = ft_mod,
  spl_mod = spl_mod,
  info = c(
    title = title,
    ft_mod = wrap_long_lines(
      "Associate genetic features with calculated module (a feature belongs to
      a module)"
    ),
    spl_mod = "Sample eigengen value (eigengen _expression profile_)",
    source = "UMB-BESD"
```

```
  ),
  dir = cfg_prog$paths$grh,
  basenm = paste0(params$basenm, "_", oid),
  rdata = TRUE
)
```

```
RData output (binary: ../tlg/graph/t2d_11_16.RData
```

```
Excel output: ../tlg/graph/t2d_11_16.xlsx
```

6.0.6 Legacy

6.0.6.1 Data preparation

6.1 Session Informations

```
sessioninfo::session_info()
```

7 GSEA / DE Results

Program 13

Francois Collin
2022-12-12

```
cfg_prog <- yaml::read_yaml("_prog.yml")
```

The Gene Set Enrichment Analysis (GSEA) aimed at transforming the interpretation of the differential expression of individual genes to a collection (or set) of genes enriched in differentially expressed genes. The set of genes being associated with identified functions, processes, molecules, diseases, pathways, etc, it offers a higher level of interpretation for the differential expression analysis.

Two sources of gene sets were used: Gene Ontology Terms (GO). The analysis will be extended to the Kyoto Encyclopedia of Genes and Genomes (KEGG).

The analysis was run on all mRNA Differential Analysis contrasts using R and the package `liger` (Fan 2019). The results focused on enriched gene sets with indication of up or down regulation while reporting the proportion of genes being found differentially expressed in each gene set.

Note: 221107, the gsea analysis is replicable.

```
devtools::load_all("src/pkg/dbs.data")
```

```
i Loading dbs.data
```

```
devtools::load_all("src/pkg/latarnia.utils")
```

```
i Loading latarnia.utils
Loading required package: grid

Loading required package: shiny
```

```
knitr::opts_chunk$set(results = cfg_prog$knitr$results)
library(assertthat)
library(ggplot2)
source("R/ngs.R")
source("R/inches.R")
source("R/export_xl.R")
```

7.0.1 Gene info

```
out_file <- "../data/dta_13_02_gene_info.RData"
```

```
# GO needs entrezgene_id
gene_info <- biomaRt::listEnsemblArchives() |>
  subset(date == "Jul 2019", select = "url") |>
  unlist(use.names = FALSE) |>
  biomaRt::useMart(
    biomart = "ensembl",
    dataset = "hsapiens_gene_ensembl",
    host = _
  ) |>
  # the biomaRt::getBM does not work well inside lapply (I don't know why)
  # + but also more efficient to get all genes data once instead of
  # + repeating the request for overlapping genes.
  biomaRt::getBM(
    mart = _,
    attributes = c(
      "ensembl_gene_id",
      "ensembl_gene_id_version",
      "external_gene_name",
      "entrezgene_id",
      "entrezgene_accession",
      "chromosome_name"
    ),
    filters = "ensembl_gene_id",
    values = rownames(dbs.data::mrna_raw),
    useCache = FALSE
  )

save(gene_info, file = out_file)
```

```r
load(out_file, verbose = TRUE)
```

```r
gene_info |>
  head() |>
  knitr::kable()
```

```r
gsets <- as.list(org.Hs.eg.db::org.Hs.egGO2EG)
```

7.0.2 GSEA preparation (helper functions)

```r
#' Abbreviate Description
#'
#' @param x (`character`)
#' @param n_char (`numeric`)
#' @export
#'
desc_abbreviate <- function(x, n_char = 50) {
  need_short <- nchar(as.character(x)) > n_char
  x[need_short] <- paste(
    substr(
      x = x[need_short],
      start = 1,
      stop = n_char
    ), "..."
  )
  x
}

## helper_functions ----

#' Geneset Preparation
#'
#' Set a list of genesets including at least three genes from a genepool.
#'
#' @param gsets (`list`)\cr a named list of genes.
#' @param universe (`character`)\cr what range of genes to consider.
#' @param control (`list`)\cr set the number of cores to run and
```

```r
#'   the minimum number of genes is returned gene sets
#' @export
#' @details
#' Restrict each gene set to the genes present in the gene background.
#' Restrict gene sets including at least 3 genes.
#'
list_genesets <- function(gsets,
                          universe,
                          control = list(
                            mingeneinset_gsea = 5,
                            minpctgeneinsetgsea = 20,
                            n_cores = parallel::detectCores()
                          )) {

  universe <- as.character(universe)
  universe <- unique(universe)

  # Restrict gsets to universe.
  gsets <- parallel::mclapply(
    X = gsets,
    universe = universe,
    FUN = function(x, universe) {
      y <- unique(x[as.character(x) %in% universe])
      attr(y, "Geneset size (nsub)") <- length(x)
      attr(y, "Geneset x Universe intersection size (n)") <- length(y)
      attr(y, "Geneset x Universe intersection size (pct)") <-
        attr(y, "Geneset x Universe intersection size (n)") /
        attr(y, "Geneset size (nsub)") * 100
      y
    },
    mc.cores = control$n_cores
  )

  gsets_length <- parallel::mclapply(
    gsets,
    FUN = attr, which = "Geneset x Universe intersection size (n)",
    mc.cores = control$n_cores
  )
  gsets <- gsets[unlist(gsets_length) >= unlist(control$mingeneinset_gsea)]

  gsets_pct <- parallel::mclapply(
```

```
    gsets,
    FUN = attr, which = "Geneset x Universe intersection size (pct)",
    mc.cores = control$n_cores
  )
  gsets <- gsets[unlist(gsets_pct) >= unlist(control$minpctgeneinsetgsea)]

  # Probably to be removed, but requires impact assessment:
  gsets <- parallel::mclapply(
    gsets,
    FUN = function(x) {
      attr(x, "genes in universe") <- length(x)
      x
    },
    mc.cores = control$n_cores
  )
  attr(gsets, "control") <- control
  gsets
}

#' BESD: GSEA
#'
#' Optimized Gene Set Enrichment Analysis for BESD.
#'
#' @param lfc (`numeric`)\cr Log2-Fold Change.
#' @param nm (`character`)\cr gene name, same length as `lfc`.
#' @param gset (`list`)\cr named list of genesets, prepared with
#'   `list_genesets`.
#' @param control (`list`)\cr list of settings as provided by `control_rna()`.
#'
#' @export
#' @details
#' Interpretation relies on:
#' <https://cran.r-project.org/web/packages/liger/vignettes/interpreting.pdf>
#'
besd_gsea_analysis <- function(lfc,
                               nm,
                               gset,
                               control = list(
                                 n_cores = parallel::detectCores(),
                                 n_rand_gsea = 1e4
                               )) {
```

```
  time_start <- Sys.time()
  values <- setNames(object = lfc, nm = nm)
  y <- lapply(
    X = gset,
    values = values,
    FUN = function(x, values = values)
      liger::gsea(
        values = values,
        geneset = x,
        mc.cores = control$n_cores,
        plot = FALSE,
        n.rand = control$n_rand_gsea,
        return.details = TRUE
      )
  )
  y <- lapply(y, t)
  y <- lapply(y, as.data.frame)
  y <- do.call(rbind, y)
  y <- cbind(
    gset = names(gset),
    y
  )

  time_end <- Sys.time();
  attr(y, "gsea_time") <- list(
    start = time_start,
    end = time_end,
    duration = time_end - time_start
  )
  attr(y, "controls") <- control
  attr(y, "gene_lfc") <- values
  y

}

#' Labels
#'
#' Get and set labels.
#' @name label
#'
NULL
```

```r
#' @describeIn label set label.
#' @param x (any)\cr any object to attribute a label to.
#' @param value (any)\cr the information used as a label for `x`.
#' @export
#' @examples
#' a <- 0
#' label(a) <- "A value of interest"
#'
`label<-` <- function(x, value) {
  attr(x, "label") <- value
  x
}

#' @describeIn label get label.
#' @export
#' @examples
#' label(a)
#'
label <- function(x) attr(x, "label")

#' GO Description
#'
#' For genesets.
#'
#' @param id (`character`)\cr the GO ID.
#' @param de_bg (`data.frame`)\cr includes `padj` and `entrezgene_id`
#'   columns.
#' @param go_desc (`list`)\cr gene set description as provided by
#'   [GO.db::GOTERM].
#' @param go_compo (`list`)\cr gene set composition as provided by
#'   [org.Hs.eg.db::org.Hs.egGO2EG].
#'
#' @return `data.frame`
#'
#' @export
#' @note
#' Because of the selection of DE gene involved in the Gene Ratio,
#' it is possible that Gene Ratio reaches 0.
#'
describe_go <- function(id,
                        de_bg,
```

```
                        go_desc = as.list(GO.db::GOTERM),
                        go_compo = as.list(org.Hs.eg.db::org.Hs.egGO2EG)) {

assertthat::assert_that(
  "padj" %in% colnames(de_bg),
  "entrezgene_id" %in% colnames(de_bg)
)
de_bg$entrezgene_id <- as.character(de_bg$entrezgene_id)

y <- lapply(
  X = id,
  de_bg = de_bg,
  go_desc = go_desc,
  go_compo = go_compo,
  FUN = function(x, de_bg, go_desc, go_compo) {
    y <- go_desc[[x]]
    go_compo <- go_compo[[x]]
    go_compo <- go_compo[
      go_compo %in% de_bg$entrezgene_id
    ]
    de_gene_in_go <- de_bg$entrezgene_id[
      de_bg$entrezgene_id %in% go_compo &
        de_bg$padj < 0.05
    ]
    data.frame(
      source = "GO",
      id = AnnotationDbi::GOID(y),
      cat = AnnotationDbi::Ontology(y),
      short_desc = AnnotationDbi::Term(y),
      size = length(go_compo),
      count = length(de_gene_in_go),
      gene_ratio = length(de_gene_in_go) / length(go_compo)
    )
  }
)

y <- do.call(rbind, y)
label(y$id) <- "Gene Ontology term ID"
label(y$cat) <- "Gene Ontology name"
label(y$short_desc) <- "Gene Ontology short term description"
label(y$size) <- "Number of Genes in GO and background"
```

```
  label(y$count) <- "Number of DE Genes in GO and background"
  label(y$gene_ratio) <- "Count / Size"
  y
}

# <https://github.com/JEFworks/liger/blob/master/vignettes/gsea.pdf>
gsea_wrapper <- function(df,
                         gs = gs,
                         gi = gene_info) {

  message("names of first gs: ", names(head(gs)))

  # Need the Entrez gene id
  gi <- gi[!is.na(gi$entrezgene_id), ]

  assertthat::assert_that(is(df, "data.frame"))
  message(unique(df$ctrs))

  res <- list()
  res$lfc <- merge(df, gi, by.x = "feature", by.y = "ensembl_gene_id")

  message("dim res$df: ", paste(dim(df), collapse = " x "))
  assertthat::assert_that(
    all(!is.na(res$entrezgene_id)),
    all(table(res$entrezgene_id) == 1L)
  )

  gs <- list_genesets(gsets = gs, universe = res$lfc$entrezgene_id)
  gc(reset = TRUE)
  message("Number of genesets: ", length(gs))

  res$gsea <- besd_gsea_analysis(
    lfc = res$lfc$log2FoldChange,
    nm = res$lfc$entrezgene_id,
    gset = gs
  )
  gc(reset = TRUE)

  res$gsea <- cbind(
    res$gsea,
    data.frame(
```

```
      padj = p.adjust(res$gsea$p.val, method = "BH"),
      regulation = ifelse(
        res$gsea$edge.value > 0,
        yes = "up-regulated",
        no = "down-regulated"
      ),
      representation =  ifelse(
        res$gsea$edge.score * res$gsea$edge.value > 0,
        yes = "enrichment",
        no = "depletion"
      )
    ),
    describe_go(id = res$gsea$gset, de_bg = res$lfc)
  )

  res
}
```

7.0.3 GSEA/Intervention Induced DE by Diabetes Group

```
out_file <- "../data/dta_13_01.RData"
```

```
result <- "../tlg/graph/fig_07_11.xlsx" |>
  openxlsx::read.xlsx(sheet = "fig_07_11", startRow = 2, rowNames = TRUE) |>
  (\(df) split(df, f = df["ctrs"]))() |>
  lapply(gs = gsets, gsea_wrapper)

save(result, file = out_file)
```

```
load(out_file, verbose = TRUE)
```

```
result |>
  lapply(\(x) head(x$gsea))
result |>
  lapply(\(x) dim(x$gsea))
```

```
contrast <- "chg NGT"
de_name <- " following Intervention"
```

```r
dta <- result[[contrast]]$gsea |>
  within({
    term <- short_desc
    short_desc <- desc_abbreviate(short_desc)
    short_desc <- factor(short_desc, unique(short_desc[order(gene_ratio)]))
  })

dta_sel <- dta |>
  subset(representation == "enrichment") |>
  (\(df) {
    df <- if (sum(df$padj < 0.05 & df$gene_ratio > 0.2) < 5) {
      df <- y[order(df$padj), ]
      df[seq_len(min(c(nrow(df), 5))), ]
    } else {
      df[df$padj < 0.05 & df$gene_ratio > 0.2, ]
    }
  }) ()

library(ggplot2)
gg <- ggplot(
  dta_sel,
  aes_string(
    x = "gene_ratio",
    color = "padj",
    y = "short_desc",
    size = "count",
    shape = "regulation"
  )
) +
  geom_point() +
  scale_colour_gradient(low = "black", high = "gray80") +
  scale_size(breaks = c(10, 50, 100, 500, 1000)) +
  scale_shape_manual(values = c(1, 19)) +
  guides(
    colour = guide_colourbar(title = "P.adj:"),
    shape = guide_legend(title = "Regulation: "),
    size = guide_legend(title = "Count: ")
  ) +
  coord_cartesian(clip = "off") +
  facet_grid(cat ~ ., scales = "free", space = "free") +
  labs(title = paste0("[mRNA] ", contrast, de_name)) +
```

```
  xlab("Gene Ratio") +
  ylab("Gene Ratio") +
  theme_minimal() +
  theme(
    axis.title.y = element_blank(),
    legend.position = "right",
    legend.box = "vertical",
    legend.text = element_text(size = 8, hjust = 0),
    legend.box.just = "left",
    panel.border = element_rect(fill = NA),
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )
```

```
p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 13.1",
      strwrap(
        "Bubble plot - Gene ontology genesets enriched with gene differentially
        expressed after exercise intervention in NGT group ",
        width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(gg, height = inches(5), width = inches(6)) |>
  add_footer(
    c(
      "Program t2d_13_gea / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )

export_as(
  p,
  file = file.path(cfg_prog$paths$grh, "fig_13_01.pdf"),
```

```
    file_graph_alone = file.path(cfg_prog$paths$grh, "fig_13_01_af.pdf")
  )
```

```
[log] output saved as: ../tlg/graph/fig_13_01.pdf
```

```
[log] output saved as: ../tlg/graph/fig_13_01_af.pdf (annot. free)
```

```
  export_xl(
    fig_13_01 = dta,
    info = c(
      title = paste("Geneset enrichment analysis", contrast, de_name),
      source = "UMB-BESD"
    ),
    dir = cfg_prog$paths$grh,
    basenm = "fig_13_01"
  )
```

```
Excel output: ../tlg/graph/fig_13_01.xlsx
```

```
  show_slate(p)
```

```
  rm(gg, p, dta, dta_sel, contrast, de_name)
```

```
  contrast <- "chg T2D"
  de_name <- " following Intervention"
```

```
  dta <- result[[contrast]]$gsea |>
    within({
      term <- short_desc
      short_desc <- desc_abbreviate(short_desc)
      short_desc <- factor(short_desc, unique(short_desc[order(gene_ratio)]))
    })

  dta_sel <- dta |>
    subset(representation == "enrichment") |>
    (\(df) {
      df <- if (sum(df$padj < 0.05 & df$gene_ratio > 0.2) < 5) {
```

```
      df <- y[order(df$padj), ]
      df[seq_len(min(c(nrow(df), 5))), ]
    } else {
      df[df$padj < 0.05 & df$gene_ratio > 0.2, ]
    }
  }) ()

library(ggplot2)
gg <- ggplot(
  dta_sel,
  aes_string(
    x = "gene_ratio",
    color = "padj",
    y = "short_desc",
    size = "count",
    shape = "regulation"
  )
) +
  geom_point() +
  scale_colour_gradient(low = "black", high = "gray80") +
  scale_size(breaks = c(10, 50, 100, 500, 1000)) +
  scale_shape_manual(values = c(1, 19)) +
  guides(
    colour = guide_colourbar(title = "P.adj:"),
    shape = guide_legend(title = "Regulation: "),
    size = guide_legend(title = "Count: ")
  ) +
  coord_cartesian(clip = "off") +
  facet_grid(cat ~ ., scales = "free", space = "free") +
  labs(title = paste0("[mRNA] ", contrast, de_name)) +
  xlab("Gene Ratio") +
  ylab("Gene Ratio") +
  theme_minimal() +
  theme(
    axis.title.y = element_blank(),
    legend.position = "right",
    legend.box = "vertical",
    legend.text = element_text(size = 8, hjust = 0),
    legend.box.just = "left",
    panel.border = element_rect(fill = NA),
    text = element_text(size = 7),
```

```
    title =  element_text(size = 7),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )
```

```
p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 13.2",
      strwrap(
        "Bubble plot - Gene ontology genesets enriched with gene differentially
        expressed after exercise intervention in T2D group ",
        width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(gg, height = inches(5), width = inches(6)) |>
  add_footer(
    c(
      "Program t2d_13_gea / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )

export_as(
  p,
  file = file.path(cfg_prog$paths$grh, "fig_13_02.pdf"),
  file_graph_alone = file.path(cfg_prog$paths$grh, "fig_13_02_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_13_02.pdf
```

```
[log] output saved as: ../tlg/graph/fig_13_02_af.pdf (annot. free)
```

```
export_xl(
  fig_13_02 = dta,
  info = c(
```

```
    title = paste("Geneset enrichment analysis", contrast, de_name),
    source = "UMB-BESD"
  ),
  dir = cfg_prog$paths$grh,
  basenm = "fig_13_02"
)
```

Excel output: ../tlg/graph/fig_13_02.xlsx

```
show_slate(p)
```

```
rm(gg, p, dta, dta_sel, contrast, de_name)
```

7.0.4 GSEA/DE between Diabetes Group at Baseline

```
out_file <- "../data/dta_13_04.RData"
```

```
# disregard the contrast IGT vs NGT because of the very low level of
# differential expression.
result <- "../tlg/graph/fig_07_03.xlsx" |>
  openxlsx::read.xlsx(sheet = "fig_07_03", startRow = 2, rowNames = TRUE) |>
  subset(ctrs %in% c("T2D vs IGT", "T2D vs NGT")) |>
  (\(df) split(df, f = df["ctrs"]))() |>
  lapply(gs = gsets, gsea_wrapper)

save(result, file = out_file)
```

```
load(out_file, verbose = TRUE)
```

```
result |>
  lapply(\(x) head(x$gsea))
result |>
  lapply(\(x) dim(x$gsea))
```

```
contrast <- "T2D vs NGT"
de_name <- " at Baseline"
```

```r
dta <- result[[contrast]]$gsea |>
  within({
    term <- short_desc
    short_desc <- desc_abbreviate(short_desc)
    short_desc <- factor(short_desc, unique(short_desc[order(gene_ratio)]))
  })

dta_sel <- dta |>
  subset(representation == "enrichment") |>
  (\(df) {
    df <- if (sum(df$padj < 0.05 & df$gene_ratio > 0.2) < 5) {
      df <- y[order(df$padj), ]
      df[seq_len(min(c(nrow(df), 5))), ]
    } else {
      df[df$padj < 0.05 & df$gene_ratio > 0.2, ]
    }
  }) ()

library(ggplot2)
gg <- ggplot(
  dta_sel,
  aes_string(
    x = "gene_ratio",
    color = "padj",
    y = "short_desc",
    size = "count",
    shape = "regulation"
  )
) +
  geom_point() +
  scale_colour_gradient(low = "black", high = "gray80") +
  scale_size(breaks = c(10, 50, 100, 500, 1000)) +
  scale_shape_manual(values = c(1, 19)) +
  guides(
    colour = guide_colourbar(title = "P.adj:"),
    shape = guide_legend(title = "Regulation: "),
    size = guide_legend(title = "Count: ")
  ) +
  coord_cartesian(clip = "off") +
  facet_grid(cat ~ ., scales = "free", space = "free") +
  labs(title = paste0("[mRNA] ", contrast, de_name)) +
```

```
  xlab("Gene Ratio") +
  ylab("Gene Ratio") +
  theme_minimal() +
  theme(
    axis.title.y = element_blank(),
    legend.position = "right",
    legend.box = "vertical",
    legend.text = element_text(size = 8, hjust = 0),
    legend.box.just = "left",
    panel.border = element_rect(fill = NA),
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )
```

```
p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
    c(
      "Figure 13.4",
      strwrap(
        "Bubble plot - Gene ontology genesets enriched with gene differentially
        expressed in T2D group compared to NGT group at Baseline",
        width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(gg, height = inches(5), width = inches(6)) |>
  add_footer(
    c(
      "Program t2d_13_gea / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )

export_as(
  p,
  file = file.path(cfg_prog$paths$grh, "fig_13_04.pdf"),
```

```
    file_graph_alone = file.path(cfg_prog$paths$grh, "fig_13_04_af.pdf")
  )
```

```
[log] output saved as: ../tlg/graph/fig_13_04.pdf
```

```
[log] output saved as: ../tlg/graph/fig_13_04_af.pdf (annot. free)
```

```
  export_xl(
    fig_13_04 = dta,
    info = c(
      title = paste("Geneset enrichment analysis", contrast, de_name),
      source = "UMB-BESD"
    ),
    dir = cfg_prog$paths$grh,
    basenm = "fig_13_04"
  )
```

```
Excel output: ../tlg/graph/fig_13_04.xlsx
```

```
  show_slate(p)
```

```
  rm(gg, p, dta, dta_sel, contrast, de_name)
```

7.0.5 GSEA/DE between Diabetes Group at Month 3

```
  out_file <- "../data/dta_13_03.RData"
```

```
  # disregard the contrast IGT vs NGT because of the very low level of
  # differential expression.
  result <- "../tlg/graph/fig_07_07.xlsx" |>
    openxlsx::read.xlsx(sheet = "fig_07_07", startRow = 2, rowNames = TRUE) |>
    subset(ctrs %in% c("T2D vs IGT", "T2D vs NGT")) |>
    (\(df) split(df, f = df["ctrs"]))() |>
    lapply(gs = gsets, gsea_wrapper)

  save(result, file = out_file)
```

```r
export_xl(
  fig_07_03 = dds_2_est,
  info = c(
    title =
      "mRNA at baseline - differential expression by diabetes group",
    source = "UMB-BESD"
  ),
  dir = cfg_prog$paths$grh,
  basenm = "fig_07_03"
)
```

```r
load(out_file, verbose = TRUE)
```

```r
result |>
  lapply(\(x) head(x$gsea))
result |>
  lapply(\(x) dim(x$gsea))
```

```r
contrast <- "T2D vs NGT"
de_name <- "After Intervention"
```

```r
dta <- result[[contrast]]$gsea |>
  within({
    term <- short_desc
    short_desc <- desc_abbreviate(short_desc)
    short_desc <- factor(short_desc, unique(short_desc[order(gene_ratio)]))
  })

dta_sel <- dta |>
  subset(representation == "enrichment") |>
  (\(df) {
    df <- if (sum(df$padj < 0.05 & df$gene_ratio > 0.2) < 5) {
      df <- y[order(df$padj), ]
      df[seq_len(min(c(nrow(df), 5))), ]
    } else {
      df[df$padj < 0.05 & df$gene_ratio > 0.2, ]
    }
  }) ()

library(ggplot2)
```

```r
gg <- ggplot(
  dta_sel,
  aes_string(
    x = "gene_ratio",
    color = "padj",
    y = "short_desc",
    size = "count",
    shape = "regulation"
  )
) +
  geom_point() +
  scale_colour_gradient(low = "black", high = "gray80") +
  scale_size(breaks = c(10, 50, 100, 500, 1000)) +
  scale_shape_manual(values = c(1, 19)) +
  guides(
    colour = guide_colourbar(title = "P.adj:"),
    shape = guide_legend(title = "Regulation: "),
    size = guide_legend(title = "Count: ")
  ) +
  coord_cartesian(clip = "off") +
  facet_grid(cat ~ ., scales = "free", space = "free") +
  labs(title = paste0("[mRNA] ", contrast, de_name)) +
  xlab("Gene Ratio") +
  ylab("Gene Ratio") +
  theme_minimal() +
  theme(
    axis.title.y = element_blank(),
    legend.position = "right",
    legend.box = "vertical",
    legend.text = element_text(size = 8, hjust = 0),
    legend.box.just = "left",
    panel.border = element_rect(fill = NA),
    text = element_text(size = 7),
    title =  element_text(size = 7),
    legend.text.align = 0,
    panel.grid.minor = element_blank()
  )
```

```r
p <- clean_slate() |>
  add_header(c("FCA Collin", "UMB BESD"), c("Confidential", "Draft")) |>
  add_title(
```

```
    c(
      "Figure 13.3",
      strwrap(
        "Bubble plot - Gene ontology genesets enriched with gene differentially
        expressed in T2D group compared to NGT group at baseline",
        width = 80
      ),
      "Analysis Set: Full Analysis Set"
    )
  ) |>
  add_figure(gg, height = inches(5), width = inches(6)) |>
  add_footer(
    c(
      "Program t2d_13_gea / Env ayup_dbs:v0.1.0-alpha",
      format(Sys.time(), format = "%Y-%m-%d %H:%M (%Z)")
    ),
    cfg_prog$version
  )

export_as(
  p,
  file = file.path(cfg_prog$paths$grh, "fig_13_03.pdf"),
  file_graph_alone = file.path(cfg_prog$paths$grh, "fig_13_03_af.pdf")
)
```

```
[log] output saved as: ../tlg/graph/fig_13_03.pdf
```

```
[log] output saved as: ../tlg/graph/fig_13_03_af.pdf (annot. free)
```

```
export_xl(
  fig_13_03 = dta,
  info = c(
    title = paste("Geneset enrichment analysis", contrast, de_name),
    source = "UMB-BESD"
  ),
  dir = cfg_prog$paths$grh,
  basenm = "fig_13_03"
)
```

```
Excel output: ../tlg/graph/fig_13_03.xlsx
```

```
show_slate(p)
```

```
rm(gg, p, dta, dta_sel, contrast, de_name)
```

7.1 Session Informations

```
sessioninfo::session_info()
```

References

Committee for Medicinal Products for Human Use (CHMP). 2015. "Guideline on Adjustment for Baseline Covariates in Clinical Trials." European Medicines Agency. https://www.ema.europa.eu/en/documents/scientific-guideline/guideline-adjustment-baseline-covariates-clinical-trials_en.pdf.

COMMITTEE FOR PROPRIETARY MEDICINAL PRODUCTS. 2003. "POINTS TO CONSIDER ON ADJUSTMENT FOR BASELINE COVARIATES." The European Agency for the Evaluation of Medicinal Products Evaluation of Medicines for Human Use. https://www.ema.europa.eu/en/documents/scientific-guideline/points-consider-adjustment-baseline-covariates_en.pdf.

Dündar, Friederike, Luce Skrabanek, and Paul Zumbo. 2018. "Introduction to Differential Gene Expression Analysis Using RNA-Seq." Internet.

Fan, Jean. 2019. *Differential Pathway Analysis*. Edited by GC Yuan. Springer.

Fox, John, and Sanford Weisberg. 2019. *An R Companion to Applied Regression*. Third. Thousand Oaks CA: Sage. https://socialsciences.mcmaster.ca/jfox/Books/Companion/.

Gohel, David. 2022. *Flextable: Functions for Tabular Reporting*.

Josse, Julie, and François Husson. 2016. "missMDA: A Package for Handling Missing Values in Multivariate Data Analysis." *Journal of Statistical Software* 70 (1): 1–31. https://doi.org/10.18637/jss.v070.i01.

Langfelder, Peter, and Steve Horvath. 2008. "WGCNA: An r Package for Weighted Correlation Network Analysis." *BMC Bioinformatics*, no. 1: 559. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-9-559.

———. 2012. "Fast R Functions for Robust Correlations and Hierarchical Clustering." *Journal of Statistical Software* 46 (11): 1–17. https://www.jstatsoft.org/v46/i11/.

Lenth, Russell V. 2022. *Emmeans: Estimated Marginal Means, Aka Least-Squares Means*. https://github.com/rvlenth/emmeans.

Liu, Guanghan F, Kaifeng Lu, Robin Mogg, Madhuja Mallick, and Devan V Mehrotra. 2009. "Should Baseline Be a Covariate or Dependent Variable in Analyses of Change from Baseline in Clinical Trials?" *Statistics in Medicine* 28 (20): 2509–30. https://doi.org/10.1002/sim.3639.

Love, Michael I., Wolfgang Huber, and Simon Anders. 2014. "Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2." *Genome Biology* 15: 550. https://doi.org/10.1186/s13059-014-0550-8.

Niemira, Magdalena, Francois Collin, Anna Szalkowska, Agnieszka Bielska, Karolina Chwialkowska, Joanna Reszec, Jacek Niklinski, Miroslaw Kwasniewski, and Adam Kretowski. 2020. "Molecular Signature of Subtypes of Non-Small-Cell Lung Cancer

by Large-Scale Transcriptional Profiling: Identification of Key Modules and Genes by Weighted Gene Co-Expression Network Analysis (WGCNA)." *Cancers* 12 (1): 37. https://doi.org/10.3390/cancers12010037.

O'Connell, Nathaniel S, Lin Dai, Yunyun Jiang, Jaime L Speiser, Ralph Ward, Wei Wei, Rachel Carroll, and Mulugeta Gebregziabher. 2017. "Methods for Analysis of Pre-Post Data in Clinical Research: A Comparison of Five Common Methods." *Journal of Biometrics & Biostatistics* 8 (1): 1. https://doi.org/10.4172/2155-6180.1000334.

R Core Team. 2022. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Ramos, Marcel, Lucas Schiffer, Angela Re, Rimsha Azhar, Azfar Basunia, Carmen Rodriguez Cabrera, Tiffany Chan, et al. 2017. "Software for the Integration of Multi-Omics Experiments in Bioconductor." *Cancer Research* 77(21); e39-42.

Van Breukelen, G. 2006. "ANCOVA Versus Change from Baseline: More Power Inrandomized Studies, More Bias in Nonrandomized Studies." *Journal of ClinicalEpidemiology*, 59–920. https://doi.org/10.1016/j.jclinepi.2006.02.007.

Vickers, Andrew J. 2001. "The Use of Percentage Change from Baseline as an Outcome in a Controlled Trial Is Statistically Inefficient: A Simulation Study." *BMC Medical Research Methodology* 1 (1): 1–4. https://doi.org/10.1186/1471-2288-1-6.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, and Maximilian Girlich. 2022. *Tidyr: Tidy Messy Dataversion*.

Wilke, Claus O. 2020. *Cowplot: Streamlined Plot Theme and Plot Annotations for 'Ggplot2'*. https://wilkelab.org/cowplot/.

Zhang, Shiyuan, James Paul, Manyat Nantha-Aree, Norman Buckley, Uswa Shahzad, Ji Cheng, Justin DeBeer, et al. 2014. "Empirical Comparison of Four Baseline Covariate Adjustment Methods in Analysis of Continuous Outcomes in Randomized Controlled Trials." *Clinical Epidemiology* 6: 227. https://doi.org/10.2147/clep.s56554.