LIMMA_script_for_DEGs_complexHeatmap.R

t

2024-11-18

```
# HEADER ####
#
# Version: 2024-11-18
#
# Figure 3: Vocano plot, Heatmap
#
# Suppl.Fig.S2
#
#
#
# SETUP ####

Sys.setenv(lang = "en_US")
```

Install required packages if missing ————————————————–

```
# Package names from CRAN
packs <- c("limma", "RColorBrewer", "ggrepel", "NMF", "pheatmap",
           "dplyr", "circlize",
           "tibble", "UpSetR",
           "ggplot2", "gridExtra", "grid", "flextable", "svglite")

# Install packages not yet installed
installed_packages <- packs %in% rownames(installed.packages())
if (any(installed_packages == FALSE)) {
  install.packages(packs[!installed_packages])
}

# Package names from BioConductor
bcpacks <- c("ComplexHeatmap")
# Install BC-packages not yet installed
installed_packages <- bcpacks %in% rownames(installed.packages())
if (any(installed_packages == FALSE)) {
  if (!require("BiocManager", quietly = TRUE)) {
    install.packages("BiocManager")}
  BiocManager::install(bcpacks[!installed_packages])
}
```

Load required packages ————————————————–

```
invisible(lapply(packs, library, character.only = TRUE)) # CRAN
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'NMF' was built under R version 4.4.2
```

```
## Loading required package: registry
```

```
## Loading required package: rngtools
```

```
## Warning: package 'rngtools' was built under R version 4.4.2

## Loading required package: cluster

## NMF - BioConductor layer [OK] | Shared memory capabilities [NO: windows] | Cores 2/2

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

##
## Attaching package: 'dplyr'

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

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

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

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

## ========================================
## circlize version 0.4.16
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: https://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
##   in R. Bioinformatics 2014.
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(circlize))
## ========================================

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

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

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

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

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

## The following object is masked from 'package:circlize':
##
##     fontsize

## The following object is masked from 'package:BiocGenerics':
##
##     width

invisible(lapply(bcpacks, library, character.only = TRUE)) # BioConductor

## ========================================
## ComplexHeatmap version 2.20.0
## Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
## Github page: https://github.com/jokergoo/ComplexHeatmap
## Documentation: http://jokergoo.github.io/ComplexHeatmap-reference
##
## If you use it in published research, please cite either one:
## - Gu, Z. Complex Heatmap Visualization. iMeta 2022.
## - Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
##     genomic data. Bioinformatics 2016.
##
##
## The new InteractiveComplexHeatmap package can directly export static
## complex heatmaps into an interactive Shiny app with zero effort. Have a try!
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(ComplexHeatmap))
## ========================================
## ! pheatmap() has been masked by ComplexHeatmap::pheatmap(). Most of the arguments
##    in the original pheatmap() are identically supported in the new function. You
##    can still use the original function by explicitly calling pheatmap::pheatmap().

##
## Attaching package: 'ComplexHeatmap'

## The following object is masked from 'package:pheatmap':
##
##     pheatmap
```

Create output directory ——————————————-

```
dir.create("./out")
```

```r
# IMPORT ####

counts <- read.delim("Paired_normalized_HTG_data.txt", stringsAsFactors = FALSE)
sampleinfo <- read.delim("SampleInfo_paired_normalized_HTG_data.txt", stringsAsFactors =
TRUE)
geneswo15 <-
c('BTG2','CYR61','DUSP1','EGR1','FOS','GADD45B','JUN','JUNB','NR4A1','PER1','RASD1','RGS2
','SERPINE1','SLC2A3','SPRY1')

row.names(counts) <- counts$pseudoID
counts$pseudoID <- NULL


# ANALYSIS ####


# quantile normalize
  y <- normalizeQuantiles(counts, ties=TRUE)
  pid <- factor(sampleinfo$PID)


# biopsy vs resect (paired)

## groups as factor
    group <- factor (sampleinfo$tissue)

## complete model
    design <- model.matrix(~0+group)
    colnames(design)
```

```
## [1] "grouppost.Tx" "grouppre.Tx"
```

```r
    #corfit <- duplicateCorrelation(y, design, block = pid)
    #corfit$consensus
    #hist(tanh(corfit$atanh.correlations))
    #write.table(corfit$atanh.correlations, file = "./results/Biop_Res_correls.txt",
row.names = F, sep = "\t", quote = F)

## linear model using weighted least squares for each gene
    fit <- lmFit(y, design, block =pid, cor = 0.2978) #corfit$consensus.correlation) #
correlation = 0.27 based on paired data)
    head(coef(fit))
```

```
##       grouppost.Tx grouppre.Tx
## A2M      10.899809   10.638146
## ABCA2     7.057145    7.474495
## ABCA3     7.645064    8.244864
## ABCA4     5.278732    5.270271
## ABCA5     6.962359    6.848244
## ABCA9     6.376097    6.007248
```

```r
    cont.matrix_B_R <- makeContrasts(
      "B_R"=(grouppost.Tx-grouppre.Tx),  levels=design)

    fit2_BR <- contrasts.fit(fit, cont.matrix_B_R)
```

```r
    fit2_BR <- eBayes(fit2_BR)
    colnames(fit2_BR)
```

```
## [1] "B_R"
```

```r
## Table of DEGs
    top.table_BR <- topTable(fit2_BR,coef='B_R',adjust.method ="BY",
                             n=Inf, p.value=0.05, confint=0.95)
    length(which(top.table_BR$adj.P.Val <= 0.05))
```

```
## [1] 1960
```

```r
    top.table_BR$Gene <- rownames(top.table_BR)
    top.table_BR <- top.table_BR[,c("Gene", names(top.table_BR)[1:8])]

## Save DEG Table
    write.table(top.table_BR, file = "./out/DEG_Biop_Res.txt", row.names = F, sep = "\t",
quote = F)


### filter statistically sig.genes
    sig_br <- top.table_BR[top.table_BR$P.Value <= 0.05, ]
    sig_br1 <- sig_br[sig_br$logFC >= 0.58, ]
    sig_br2 <- sig_br[sig_br$logFC <= -0.58, ]
    length(which(sig_br$P.Val <= 0.05))
```

```
## [1] 1960
```

```r
    length(which(sig_br$logFC >= 0.58))
```

```
## [1] 171
```

```r
    length(which(sig_br$logFC <= -0.58))
```

```
## [1] 164
```

```r
    stat_sig_br <- data.frame(rbind(sig_br1,sig_br2) )

    adj.sig_br <- top.table_BR[top.table_BR$adj.P.Val <= 0.05, ]
    adj.sig_br1 <- adj.sig_br[adj.sig_br$logFC >= 0.58, ]
    adj.sig_br2 <- adj.sig_br[adj.sig_br$logFC <= -0.58, ]
    length(which(adj.sig_br$adj.P.Val <= 0.05))
```

```
## [1] 1960
```

```r
    length(which(adj.sig_br$logFC >= 0.58))
```

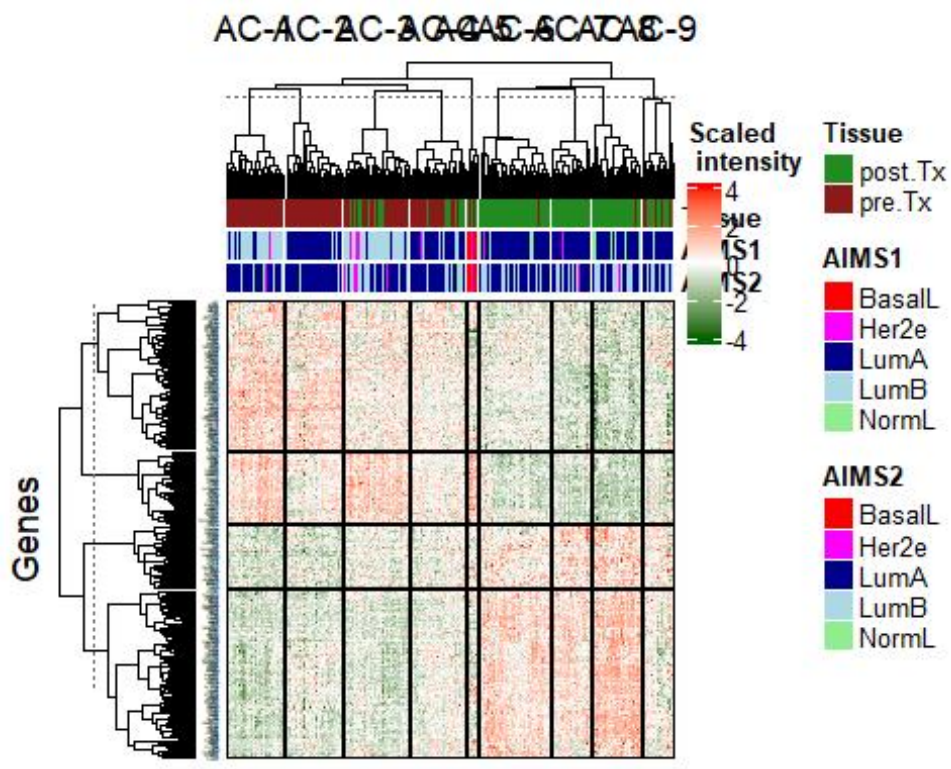
```
## [1] 171
```

```r
    length(which(adj.sig_br$logFC <= -0.58))
```

```
## [1] 164
```

```r
    adj_sig_br <- data.frame(rbind(adj.sig_br1,adj.sig_br2) )



# PLOTS ####
```

```r
## Heatmap
    data_BR <- as.matrix(y)
    col <- as.vector(colnames(data_BR))
    mat_col1 <-sampleinfo[sampleinfo$SampleName %in% col, ]
    mat_col <- data.frame(AIMS2=c(mat_col1$aims2),AIMS1=c(mat_col1$aims1),
                          Tissue=c(mat_col1$tissue))
    rownames(mat_col) <- colnames(data_BR)

    ann_colors =  list(
      AIMS1=c('BasalL'="red",'Her2e'="magenta",'LumA'='darkblue',
              'LumB'="lightblue",'NormL'="lightgreen"),
      AIMS2=c('BasalL'="red",'Her2e'="magenta",'LumA'='darkblue',
              'LumB'="lightblue",'NormL'="lightgreen"),
      Tissue=c('pre.Tx'='firebrick4','post.Tx'='forestgreen')
      )

    allgenes <-c(rownames(data_BR))
    id_br<-rownames(stat_sig_br)
    stat.sig_br <-c(id_br)

    heatmap_br <- data_BR[allgenes %in% stat.sig_br, ]

  set.seed(123)
  cols  = colorRamp2(c(-4, 0, 4), c("darkgreen", "white", "red"))
  hmpc<- ComplexHeatmap::pheatmap(
      heatmap_br,
      row_km=4,column_km = 10,
      row_gap = unit(0.25, "mm"), column_gap = unit(0.25, "mm"),
      border = TRUE, border_gp = gpar(col = "black"),
      row_km_repeats=150,  column_km_repeats=150,
      col=cols,

      scale = "row",
      clustering_distance_rows = "euclidean",
      clustering_distance_cols = "euclidean", clustering_method = "ward.D2",
      annotation_col=mat_col,
      annotation_colors=ann_colors,
      border_color = NA,

      show_colnames = F,
      show_rownames = T, row_title = "Genes",
      row_names_side = "left", row_dend_side = "left",
      fontsize = 1,
      heatmap_legend_param =list(title ="Scaled \n intensity"),
      column_title = c('AC-1', 'AC-2','AC-3','AC-4',
                       'AC-5', 'AC-6','AC-7','AC-8','AC-9'))

hmpc = draw(hmpc)
```




```
# svglite("./out/Fig3B_heatmap.svg", width = 10, height = 8, pointsize = 12)
# draw(hmpc)
# dev.off()
```

```
# extract samples and clusters
  c.dend <- column_dend(hmpc)
  ccl.list <- column_order(hmpc)  #Extract clusters (output is a list)
  lapply(ccl.list, function(x) length(x))  #check/confirm size clusters
```

```
## $`8`
## [1] 144
##
## $`7`
## [1] 141
##
## $`4`
## [1] 162
##
## $`5`
## [1] 137
##
## $`NA`
## [1] 27
##
## $`3`
## [1] 175
##
```

```
## $`2`
## [1] 98
##
## $`9`
## [1] 120
##
## $`1`
## [1] 76
```

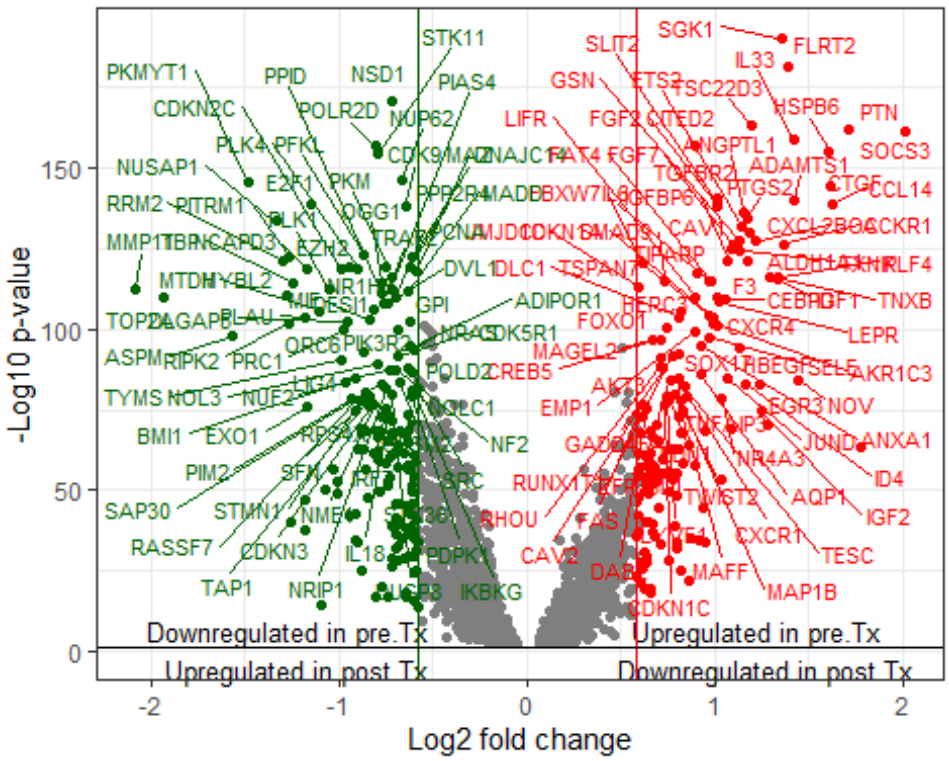
```
  for (i in 1:length(column_order(hmpc))){   if (i == 1) {
    clu <- t(t(colnames(heatmap_br[,column_order(hmpc)[[i]]])))
    out <- cbind(clu, paste("cluster", i, sep=""))
    colnames(out) <- c("sample", "Cluster")   } else {
      clu <- t(t(colnames(heatmap_br[,column_order(hmpc)[[i]]])))
      clu <- cbind(clu, paste("cluster", i, sep=""))
      out <- rbind(out, clu)   }
  }
  write.table(out, file= "./out/sample_clusters_Penelope_150rpt.txt",
              sep="\t", quote=F, row.names=FALSE)



# loop to extract genes for each cluster.
  r.dend <- row_dend(hmpc)
  rcl.list <- row_order(hmpc)  #Extract clusters (output is a list)
  lapply(rcl.list, function(x) length(x))  #check/confirm size clusters
```

```
## $`1`
## [1] 111
##
## $`2`
## [1] 53
##
## $`4`
## [1] 47
##
## $`3`
## [1] 124
```

```
  for (i in 1:length(row_order(hmpc))){  if (i == 1) {
      clu_r<- t(t(row.names(heatmap_br[row_order(hmpc)[[i]],])))
      out_r <- cbind(clu_r, paste("cluster", i, sep=""))
      colnames(out_r) <- c("coordinates", "Cluster")
    } else {
      clu_r <- t(t(row.names(heatmap_br[row_order(hmpc)[[i]],])))
      clu_r <- cbind(clu_r, paste("cluster", i, sep=""))
      out_r <- rbind(out_r, clu_r)
    }
  }
  write.table(out_r, file= "./out/gene_clusters_Penelope_150rpt.txt",
              sep="\t", quote=F, row.names=FALSE)
```

```r
## Volcano plot from Fig.3A
##  (excluding 15 genes known to be highly-inducible by preanalytical factors
##   in surgical samples).

  genessig <-c(rownames(top.table_BR))
  # for volcano plot without 15 genes affected by pre-analytical factors
  data_volcano <- top.table_BR[setdiff(rownames(top.table_BR), geneswo15),]

  logFC <- data_volcano$logFC
  P.value <- data_volcano$P.Value
  Gene <- data_volcano$Gene

  df <- data.frame(logFC, P.value, Gene)

# Labelling differentially expressed genes as UP , DOWN and NO
  df$diffexpressed <- "No"
  df$diffexpressed[df$logFC > 0.58 & df$P.value < 0.05] <- "Up"
  df$diffexpressed[df$logFC < -0.58 & df$P.value < 0.05] <- "Down"

# sort the data based on pvalue
#  df %>% arrange(P.value)

# Top 250 genes and label the genes with symbol
  Top_Hits <-  df %>% dplyr::top_n(250, wt = -log10(P.value))
  df$label = dplyr::if_else(df$Gene %in% Top_Hits$Gene,  df$Gene, "FALSE")

# create a summary table
  df$DEG =NA
  df$DEG[df$diffexpressed=="Up"] <- "Upregulated (logFC >0.58)"
  df$DEG[df$diffexpressed=="Down"] <- "Downregulated (logFC <-0.58)"

  d <- as.data.frame(table(df[,c("DEG")]))
  colnames (d) <- c("DEG (n:1960; FDR <0.05)","Total")

  dflabels <- df %>% filter(!is.na(DEG))

  p <- ggplot(df, aes(x = logFC, y = -log10(P.value), col = DEG,
                      show.legend = FALSE)) + #label=label
    geom_point() +
    theme(legend.position = "none", axis.text.x = element_text(size = 14),
          axis.text.y = element_text(size = 14) ) +

  annotate("text", label = "Downregulated in pre.Tx \n Upregulated in post.Tx",
           x = -1.25, y = 0.75, size=3.5) +          #\nin treatment",
  annotate("text", label = "Upregulated in pre.Tx \n Downregulated in post.Tx",
           x = 1.25, y = 0.75, size= 3.5) +
  labs(x = "Log2 fold change", y = "-Log10 p-value", family = "Calibri") +

  scale_color_manual(values = c("darkgreen","red", "grey")) +
  geom_vline(xintercept = c(-0.58, 0.58), col =c("darkgreen","red")) +
  geom_hline(yintercept = -log10(0.05), col ="black") +
  theme(base_size = 10) +
  geom_text_repel(data=dflabels[1:150,],
                  max.overlaps = Inf,aes(x = logFC, y = -log10(P.value),
```

```
                                        col = DEG, label=label), size=3) +

  theme_bw()+
  annotation_custom (tableGrob (d, rows=NULL,
                                theme = ttheme_default(base_size = 8)),
                     xmin=-2, xmax=-1,  ymin=255, ymax=300) +
  theme(legend.position="none") +
  coord_cartesian(clip = "off")

  p+theme(legend.position="none")
```



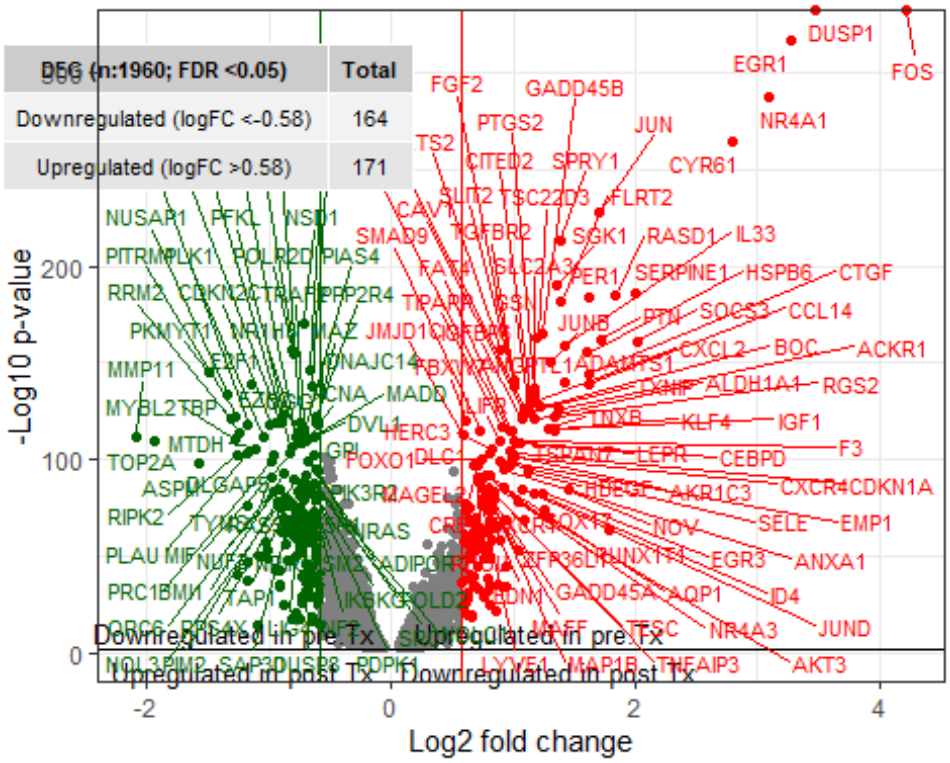

```
  svglite("./out/Fig3A_volcanoplot_wo15.svg", width = 10, height = 8)
  plot(p)
  dev.off()
```

```
## png
##   2
```

```
# Panel figure:  Figure 3A + 3B

  # Convert the heatmap to a grob object
  heatmap_grob <- grid::grid.grabExpr(draw(hmpc))

  # Save the combined figure as a PDF
  pdf("./out/combined_panel_figure3.pdf",  width = 10, height = 9)  # Convert pixels to
inches (1 inch = 72 pixels)

  # Arrange the plots in a single panel
  grid.arrange(p, heatmap_grob, ncol = 1, heights = c(3.5, NULL, 4.5))

  dev.off()
```

```
## png
##   2

# Suppl. Figure S2:

  ## Volcano plot from SFig S2 using all genes including the 15 genes
  ##  known to be highly-inducible by preanalytical factors in surgical samples


  genessig <-c(rownames(top.table_BR))
  data_volcano <- top.table_BR # ALL GENES

  logFC <- data_volcano$logFC
  P.value <- data_volcano$P.Value
  Gene <- data_volcano$Gene

  df <- data.frame(logFC, P.value, Gene)

  # Labelling differentially expressed genes as UP , DOWN and NO
  df$diffexpressed <- "No"
  df$diffexpressed[df$logFC > 0.58 & df$P.value < 0.05] <- "Up"
  df$diffexpressed[df$logFC < -0.58 & df$P.value < 0.05] <- "Down"

  # sort the data based on pvalue
  # df %>% arrange(P.value)

  # Top 250 genes and label the genes with symbol
  Top_Hits <-  df %>% dplyr::top_n(250, wt = -log10(P.value))
  df$label = dplyr::if_else(df$Gene %in% Top_Hits$Gene,  df$Gene, "FALSE")

  # create a summary table
  df$DEG =NA
  df$DEG[df$diffexpressed=="Up"] <- "Upregulated (logFC >0.58)"
  df$DEG[df$diffexpressed=="Down"] <- "Downregulated (logFC <-0.58)"

  d <- as.data.frame(table(df[,c("DEG")]))
  colnames (d) <- c("DEG (n:1960; FDR <0.05)","Total")

  dflabels <- df %>% filter(!is.na(DEG))

  p <- ggplot(df, aes(x = logFC, y = -log10(P.value), col = DEG,
                      show.legend = FALSE)) + #label=label
    geom_point() +
    theme(legend.position = "none", axis.text.x = element_text(size = 14),
          axis.text.y = element_text(size = 14) ) +

    annotate("text", label = "Downregulated in pre.Tx \n Upregulated in post.Tx",
             x = -1.25, y = 0.75, size=3.5) +          #\nin treatment",
    annotate("text", label = "Upregulated in pre.Tx \n Downregulated in post.Tx",
             x = 1.25, y = 0.75, size= 3.5) +
    labs(x = "Log2 fold change", y = "-Log10 p-value", family = "Calibri") +

    scale_color_manual(values = c("darkgreen","red", "grey")) +
    geom_vline(xintercept = c(-0.58, 0.58), col =c("darkgreen","red")) +
```

```
    geom_hline(yintercept = -log10(0.05), col ="black") +
    theme(base_size = 10) +
    geom_text_repel(data=dflabels[1:150,],
                    max.overlaps = Inf,aes(x = logFC, y = -log10(P.value),
                                           col = DEG, label=label), size=3) +


    theme_bw()+
    annotation_custom (tableGrob (d, rows=NULL,
                                  theme = ttheme_default(base_size = 8)),
                       xmin=-2, xmax=-1,  ymin=255, ymax=300) +
    theme(legend.position="none") +
    coord_cartesian(clip = "off")

  p+theme(legend.position="none")
```




```
  svglite("./out/SFigS2_volcanoplot_all335.svg", width = 10, height = 8)
  plot(p)
  dev.off()
```

```
## png
##   2
```

```
# SESSION INFO ####
  sessionInfo()
```

```
## R version 4.4.1 (2024-06-14 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 22631)
##
## Matrix products: default
##
##
## locale:
```

```
## [1] LC_COLLATE=German_Germany.utf8  LC_CTYPE=German_Germany.utf8
## [3] LC_MONETARY=German_Germany.utf8 LC_NUMERIC=C
## [5] LC_TIME=German_Germany.utf8
##
## time zone: Europe/Berlin
## tzcode source: internal
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] ComplexHeatmap_2.20.0 svglite_2.1.3         flextable_0.9.6
##  [4] gridExtra_2.3         UpSetR_1.4.0          tibble_3.2.1
##  [7] circlize_0.4.16       dplyr_1.1.4           pheatmap_1.0.12
## [10] NMF_0.28              Biobase_2.64.0        BiocGenerics_0.50.0
## [13] cluster_2.1.6         rngtools_1.5.2        registry_0.5-1
## [16] ggrepel_0.9.6         ggplot2_3.5.1         RColorBrewer_1.1-3
## [19] limma_3.60.4
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.1        gridBase_0.4-7          farver_2.1.2
##  [4] fastmap_1.2.0           fontquiver_0.2.1        digest_0.6.37
##  [7] lifecycle_1.0.4         statmod_1.5.0           magrittr_2.0.3
## [10] compiler_4.4.1          rlang_1.1.4             tools_4.4.1
## [13] utf8_1.2.4              yaml_2.3.10             data.table_1.16.0
## [16] knitr_1.48              labeling_0.4.3          askpass_1.2.0
## [19] plyr_1.8.9              xml2_1.3.6              withr_3.0.1
## [22] stats4_4.4.1            fansi_1.0.6             gdtools_0.4.0
## [25] colorspace_2.1-1        scales_1.3.0            iterators_1.0.14
## [28] cli_3.6.3               crayon_1.5.3            rmarkdown_2.28
## [31] ragg_1.3.2              generics_0.1.3          rstudioapi_0.16.0
## [34] rjson_0.2.22            reshape2_1.4.4          stringr_1.5.1
## [37] parallel_4.4.1          BiocManager_1.30.25     matrixStats_1.4.1
## [40] vctrs_0.6.5             fontBitstreamVera_0.1.1 S4Vectors_0.42.1
## [43] IRanges_2.38.1          GetoptLong_1.0.5        clue_0.3-65
## [46] systemfonts_1.1.0       foreach_1.5.2           glue_1.7.0
## [49] codetools_0.2-20        stringi_1.8.4           shape_1.4.6.1
## [52] gtable_0.3.5            munsell_0.5.1           pillar_1.9.0
## [55] htmltools_0.5.8.1       openssl_2.2.1           R6_2.5.1
## [58] textshaping_0.4.0       doParallel_1.0.17       evaluate_1.0.0
## [61] png_0.1-8               fontLiberation_0.1.0    Rcpp_1.0.13
## [64] zip_2.3.1               uuid_1.2-1              officer_0.6.6
## [67] xfun_0.47               pkgconfig_2.0.3         GlobalOptions_0.1.2
```