Results integration

2023-12-21

```
# Set the CRAN mirror
options(repos = c(CRAN = "https://cran.rstudio.com"))
install.packages("gplots")
install.packages("dplyr")
library(dplyr)
library(gplots)
library(tidyr)
```

Integrating the results of Differential Gene Expression Analysis

Loading the results of the analysis of the different datasets

```
DGE1 <- read.table("Dataset1_limma_results.txt", header = TRUE, sep = "\t")
DGE2 <- read.table("Dataset2_limma_results.txt", header = TRUE, sep = "\t")
DGE3 <- read.table("Dataset3_limma_results.txt", header = TRUE, sep = "\t")
DGE4 <- read.table("Dataset4_limma_results.txt", header = TRUE, sep = "\t")
head(DGE1)
```

```
##                  logFC  AveExpr         t      P.Value adj.P.Val         B
## 219073_s_at -0.9404248 8.169289 -6.786883 7.062747e-05 0.9999858 -2.905081
## 213802_at   -0.8614519 7.877965 -6.616723 8.610008e-05 0.9999858 -2.925224
## 203200_s_at -0.5410565 9.625066 -6.429321 1.075287e-04 0.9999858 -2.948733
## 213975_s_at  1.2691885 5.194109  6.402909 1.109890e-04 0.9999858 -2.952163
## 204320_at   -1.1461782 7.829749 -5.955744 1.923307e-04 0.9999858 -3.015029
## 37892_at    -1.3963016 8.254226 -5.636194 2.895403e-04 0.9999858 -3.066081
##                                                                        Gene
## 219073_s_at                                    oxysterol binding protein like 10
## 213802_at                                                   serine protease 12
## 203200_s_at 5-methyltetrahydrofolate-homocysteine methyltransferase reductase
## 213975_s_at                                                         lysozyme
## 204320_at                                    collagen type XI alpha 1 chain
## 37892_at                                     collagen type XI alpha 1 chain
##              Symbol EntrezID
## 219073_s_at OSBPL10   114884
## 213802_at    PRSS12     8492
## 203200_s_at    MTRR     4552
## 213975_s_at     LYZ     4069
## 204320_at   COL11A1     1301
## 37892_at    COL11A1     1301
```

```
head(DGE2)
```

```
##                     logFC   AveExpr         t      P.Value  adj.P.Val        B
## ENSG00000159674  1.077184  6.072442  8.676081 2.673478e-06 0.03323942 5.143870
## ENSG00000166257 -1.503025  5.676478 -8.262216 4.312968e-06 0.03323942 4.672593
## ENSG00000201059  4.689397 -3.025543  7.913281 6.546112e-06 0.03323942 1.983946
## ENSG00000263499 -1.794698  1.981602 -7.887603 6.753776e-06 0.03323942 4.160554
## ENSG00000114771  1.541361  6.397131  7.800815 7.509756e-06 0.03323942 4.092922
## ENSG00000184956 -3.218921  1.520790 -7.698638 8.518196e-06 0.03323942 3.777795
##                   GENENAME ensembl_gene_id ensembl_transcript_id
## ENSG00000159674     SPON2 ENSG00000159674       ENST00000290902
## ENSG00000166257     SCN3B ENSG00000166257       ENST00000527125
## ENSG00000201059 RNA5SP336 ENSG00000201059       ENST00000364189
## ENSG00000263499           ENSG00000276197       ENST00000619317
## ENSG00000114771     AADAC ENSG00000114771       ENST00000232892
## ENSG00000184956      MUC6 ENSG00000277518       ENST00000627256
##                 transcript_length gene_symbol EntrezID
## ENSG00000159674              1579       SPON2    10417
## ENSG00000166257              3289       SCN3B    55800
## ENSG00000201059               117   RNA5SP336       NA
## ENSG00000263499                90                   NA
## ENSG00000114771              1563       AADAC       13
## ENSG00000184956             14829        MUC6     4588
```

```
head(DGE3)
```

```
##        logFC  AveExpr         t      P.Value   adj.P.Val        B
## 1 -0.7757138 8.485872 -7.585632 3.127032e-07 0.003483982 6.781208
## 2 -0.5265489 7.827172 -7.880734 1.770671e-07 0.003483982 7.292687
## 3 -0.5113104 8.670438 -6.800017 1.504875e-06 0.003727136 5.352133
## 4  0.5643422 7.197586  6.298666 4.282748e-06 0.003727136 4.389659
## 5  0.5234048 9.607172  6.376655 3.631814e-06 0.003727136 4.541891
## 6 -0.3724101 6.789247 -6.554093 2.503270e-06 0.003727136 4.884825
##                gene
## 1             SSBP3
## 2 DGCR8 /// MIR1306
## 3             MMP14
## 4             DDX24
## 5             CREG1
## 6             VAMP2
```

```
head(DGE4)
```

```
##     Probe_ID DESIGN COLOR_CHANNEL      logFC   AveExpr         t      P.Value
## 1 cg23647968      I           Grn -0.3007335 2.0588751 -7.005270 2.106791e-09
## 2 cg04737885     II          Both -0.7155935 1.4505993 -6.532893 1.374686e-08
## 3 cg03197935     II          Both -0.5485005 2.0684252 -6.496843 1.585170e-08
## 4 cg09456760      I           Grn -0.2145767 1.1294714 -6.261053 4.012525e-08
## 5 cg03349251     II          Both -0.4278613 0.5681522 -5.666390 4.046752e-07
## 6 cg08092966     II          Both  0.3068525 1.6792284  5.655763 4.215249e-07
##     adj.P.Val        B   Genes   CHR      POS      PCOS PCOS_meth Control_meth
## 1 0.001020056 9.493648    RBL2 chr19 15051936 0.8423509 0.8423509    0.7993690
```

```
## 2 0.002558333 8.035692 C3orf35 chr16  4014095 0.8975500 0.8975500    0.8443069
## 3 0.002558333 7.924349  FNDC3B chr11 77885410 0.9287033 0.9287033    0.9005007
## 4 0.004856911 7.196555         chr22 51206645 0.7088972 0.7088972    0.6676127
## 5 0.030915780 5.372000   VDAC3  chr6 10832472 0.8387344 0.8387344    0.7957254
## 6 0.030915780 5.339638   ACTN1  chr8 19009591 0.8943291 0.8943291    0.9029527
##   abs_diff_meth Entrez_id
## 1    0.04298198        NA
## 2    0.05324310        NA
## 3    0.02820257        NA
## 4    0.04128453        NA
## 5    0.04300896        NA
## 6    0.00862358        NA
```

Only the columns of interest are kept, and columnnames are unified. Rows that have no gene symbol are omitted from the dataframe.

```
DGE1 <- DGE1[, c("logFC", "adj.P.Val", "Symbol")]
DGE1 <- DGE1[complete.cases(DGE1$Symbol), ]
DGE1 <- DGE1[DGE1$Symbol != "", , drop = FALSE]

DGE2$Symbol <- DGE2$gene_symbol
DGE2 <- DGE2[, c("logFC", "adj.P.Val", "Symbol")]
DGE2 <- DGE2[complete.cases(DGE2$Symbol), ]
DGE2 <- DGE2[DGE2$Symbol != "", , drop = FALSE]

DGE3$Symbol <- DGE3$gene
DGE3 <- DGE3[, c("logFC", "adj.P.Val", "Symbol")]
DGE3 <- DGE3[complete.cases(DGE3$Symbol), ]
DGE3 <- DGE3[DGE3$Symbol != "", , drop = FALSE]

DGE4$Symbol <- DGE4$Genes
DGE4 <- DGE4[, c("logFC", "adj.P.Val", "Symbol")]
DGE4 <- DGE4[complete.cases(DGE4$Symbol), ]
DGE4 <- DGE4[DGE4$Symbol != "", , drop = FALSE]

head(DGE1)
```

```
##                  logFC adj.P.Val  Symbol
## 219073_s_at -0.9404248 0.9999858 OSBPL10
## 213802_at   -0.8614519 0.9999858  PRSS12
## 203200_s_at -0.5410565 0.9999858    MTRR
## 213975_s_at  1.2691885 0.9999858     LYZ
## 204320_at   -1.1461782 0.9999858 COL11A1
## 37892_at    -1.3963016 0.9999858 COL11A1
```

```
head(DGE2)
```

```
##                     logFC  adj.P.Val    Symbol
## ENSG00000159674  1.077184 0.03323942     SPON2
## ENSG00000166257 -1.503025 0.03323942     SCN3B
## ENSG00000201059  4.689397 0.03323942 RNA5SP336
## ENSG00000114771  1.541361 0.03323942     AADAC
## ENSG00000184956 -3.218921 0.03323942      MUC6
## ENSG00000118137  1.221194 0.03394154     APOA1
```

```
head(DGE3)
```

```
##        logFC   adj.P.Val            Symbol
## 1 -0.7757138 0.003483982             SSBP3
## 2 -0.5265489 0.003483982 DGCR8 /// MIR1306
## 3 -0.5113104 0.003727136             MMP14
## 4  0.5643422 0.003727136             DDX24
## 5  0.5234048 0.003727136             CREG1
## 6 -0.3724101 0.003727136             VAMP2
```

```
head(DGE4)
```

```
##        logFC   adj.P.Val  Symbol
## 1 -0.3007335 0.001020056    RBL2
## 2 -0.7155935 0.002558333 C3orf35
## 3 -0.5485005 0.002558333  FNDC3B
## 5 -0.4278613 0.030915780   VDAC3
## 6  0.3068525 0.030915780   ACTN1
## 7 -0.2082511 0.030915780  ATP2A1
```

The third dataset showed the highest number of significantly differentially expressed genes.

```
dim(DGE3[DGE3$adj.P.Val < 0.05, ])
```

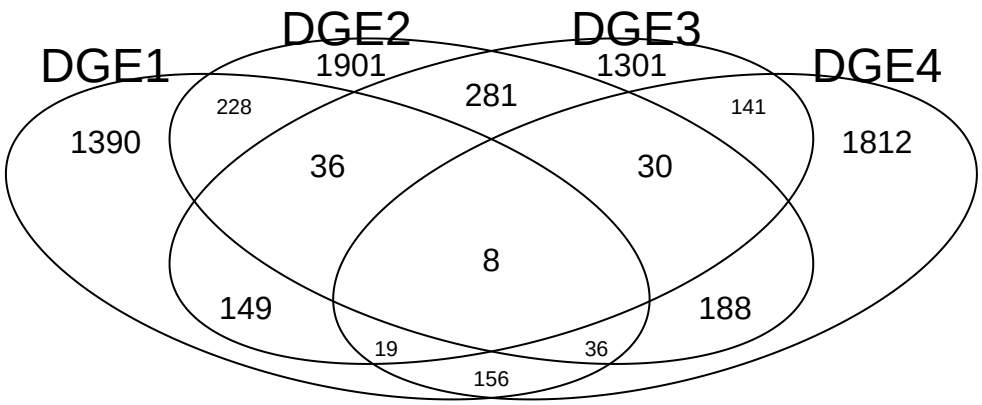
```
## [1] 5112    3
```

The first 5112 genes are selected from each dataset, from which respectively upregulated and downregulated genes are used.

```
DGE1_comp <- DGE1[1:5112, ]
DGE2_comp <- DGE2[1:5112, ]
DGE3_comp <- DGE3[1:5112, ]
DGE4_comp <- DGE4[1:5112, ]

DGE1_up <- DGE1_comp$Symbol[DGE1_comp$logFC > 0]
DGE2_up <- DGE2_comp$Symbol[DGE2_comp$logFC > 0]
DGE3_up <- DGE3_comp$Symbol[DGE3_comp$logFC > 0]
DGE4_up <- DGE4_comp$Symbol[DGE4_comp$logFC > 0]

DGE1_down <- DGE1_comp$Symbol[DGE1_comp$logFC < 0]
DGE2_down <- DGE2_comp$Symbol[DGE2_comp$logFC < 0]
DGE3_down <- DGE3_comp$Symbol[DGE3_comp$logFC < 0]
DGE4_down <- DGE4_comp$Symbol[DGE4_comp$logFC < 0]
```

Upregulated Genes Overlap from subset

```
# Extract unique symbols from each set of upregulated genes
symbols_up <- list(DGE1 = unique(DGE1_up),
                   DGE2 = unique(DGE2_up),
                   DGE3 = unique(DGE3_up),
                   DGE4 = unique(DGE4_up))

# Create Venn diagram for upregulated genes
venn <- venn(symbols_up)
```



There are 8 overlapping upregulated genes found.

```
# Extract unique symbols from each upregulated dataframe
symbols_DGE1_up <- unique(DGE1_up)
symbols_DGE2_up <- unique(DGE2_up)
symbols_DGE3_up <- unique(DGE3_up)
symbols_DGE4_up <- unique(DGE4_up)

# Find common symbols among upregulated genes
common_symbols_up <- Reduce(intersect, list(symbols_DGE1_up, symbols_DGE2_up, symbols_DGE3_up, symbols_

# Print or use common_symbols_up as needed
print(common_symbols_up)
```

```
## [1] "TGOLN2"  "CRYZ"    "SQSTM1"  "ALG5"    "BLVRA"   "ST3GAL1" "CAMLG"
## [8] "SNCA"
```

The adj.p.value and logFC of each of the genes is checked for each dataset

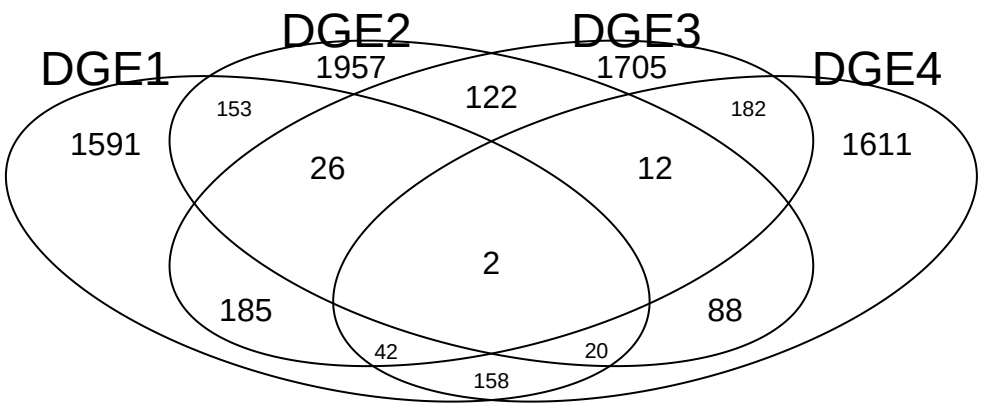
```
# Create a vector with the overlapping genes
overlapping_genes <- c("TGOLN2", "CRYZ", "SQSTM1", "ALG5", "BLVRA", "ST3GAL1", "CAMLG", "SNCA")

# Function to extract relevant columns for each dataset
extract_columns <- function(dataset, genes) {
  result <- dataset[dataset$Symbol %in% genes, c("Symbol", "adj.P.Val", "logFC")]

  # Convert "logFC" to numeric, as it might contain non-numeric values
  result$logFC <- as.numeric(as.character(result$logFC))

  colnames(result) <- c("Symbol", "adj.p.value", "logFC")

 # Keep the row with the maximum absolute logFC for each unique gene symbol
  result <- result %>%
    group_by(Symbol) %>%
    filter(logFC == max(logFC))

  return(result)
}

# Extract columns for each dataset
DGE1_subset <- extract_columns(DGE1, overlapping_genes)
DGE2_subset <- extract_columns(DGE2, overlapping_genes)
DGE3_subset <- extract_columns(DGE3, overlapping_genes)
DGE4_subset <- extract_columns(DGE4, overlapping_genes)


# Combine the four subsets into a single dataframe
combined_df <- bind_rows(
  mutate(DGE1_subset, Dataset = "Dataset 1"),
  mutate(DGE2_subset, Dataset = "Dataset 2"),
  mutate(DGE3_subset, Dataset = "Dataset 3"),
  mutate(DGE4_subset, Dataset = "Dataset 4"))

sorted_df <- combined_df %>% arrange(Symbol)

sorted_df
```

```
## # A tibble: 32 x 4
## # Groups:   Symbol [8]
##    Symbol adj.p.value logFC Dataset
##    <chr>        <dbl> <dbl> <chr>
##  1 ALG5        1.00   0.322 Dataset 1
##  2 ALG5        0.217  0.343 Dataset 2
##  3 ALG5        0.0408 0.309 Dataset 3
##  4 ALG5        0.599  0.129 Dataset 4
##  5 BLVRA       1.00   0.472 Dataset 1
##  6 BLVRA       0.193  0.455 Dataset 2
##  7 BLVRA       0.0115 0.557 Dataset 3
##  8 BLVRA       0.746  0.137 Dataset 4
##  9 CAMLG       1.00   0.230 Dataset 1
## 10 CAMLG       0.113  0.485 Dataset 2
```

```
## # i 22 more rows
```

Downregulated Genes Overlap from subset

```
# Extract unique symbols from each set of upregulated genes
symbols_down <- list(DGE1 = unique(DGE1_down),
                     DGE2 = unique(DGE2_down),
                     DGE3 = unique(DGE3_down),
                     DGE4 = unique(DGE4_down))

# Create Venn diagram for upregulated genes
venn <- venn(symbols_down)
```



There are 2 downregulated genes found.

```
# Extract unique symbols from each downregulated dataframe
symbols_DGE1_down <- unique(DGE1_down)
symbols_DGE2_down <- unique(DGE2_down)
symbols_DGE3_down <- unique(DGE3_down)
symbols_DGE4_down <- unique(DGE4_down)

# Find common symbols among downregulated genes
common_symbols_down <- Reduce(intersect, list(symbols_DGE1_down, symbols_DGE2_down, symbols_DGE3_down, s
```

```r
# Print or use common_symbols_down as needed
print(common_symbols_down)
```

```
## [1] "DEPDC1" "NTRK3"
```

The adj.p.value and logFC of each of the genes is checked for each dataset

```r
# Create a vector with the overlapping genes
overlapping_genes <- c("DEPDC1", "NTRK3")

# Function to extract relevant columns for each dataset
extract_columns <- function(dataset, genes) {
  result <- dataset[dataset$Symbol %in% genes, c("Symbol", "adj.P.Val", "logFC")]

  # Convert "logFC" to numeric, as it might contain non-numeric values
  result$logFC <- as.numeric(as.character(result$logFC))

  colnames(result) <- c("Symbol", "adj.p.value", "logFC")

 # Keep the row with the maximum absolute logFC for each unique gene symbol
  result <- result %>%
    group_by(Symbol) %>%
    filter(logFC == min(logFC))

  return(result)
}

# Extract columns for each dataset
DGE1_subset <- extract_columns(DGE1, overlapping_genes)
DGE2_subset <- extract_columns(DGE2, overlapping_genes)
DGE3_subset <- extract_columns(DGE3, overlapping_genes)
DGE4_subset <- extract_columns(DGE4, overlapping_genes)


# Combine the four subsets into a single dataframe
combined_df <- bind_rows(
  mutate(DGE1_subset, Dataset = "Dataset 1"),
  mutate(DGE2_subset, Dataset = "Dataset 2"),
  mutate(DGE3_subset, Dataset = "Dataset 3"),
  mutate(DGE4_subset, Dataset = "Dataset 4"))

sorted_df <- combined_df %>% arrange(Symbol)

sorted_df
```

```
## # A tibble: 8 x 4
## # Groups:   Symbol [2]
##   Symbol adj.p.value  logFC Dataset
##   <chr>        <dbl>  <dbl> <chr>
## 1 DEPDC1      1.00   -0.855 Dataset 1
## 2 DEPDC1      0.189  -1.10  Dataset 2
## 3 DEPDC1      0.0375 -0.295 Dataset 3
```

```
## 4 DEPDC1      0.537  -0.359 Dataset 4
## 5 NTRK3       1.00   -0.173 Dataset 1
## 6 NTRK3       0.0832 -1.08  Dataset 2
## 7 NTRK3       0.0383 -0.325 Dataset 3
## 8 NTRK3       0.665  -0.131 Dataset 4
```