

BDA_2024_MetaAnalysis_ViralEncephalitis

Duy Nguyen

2024-07-22

Research Flow of Meta-Analysis of Transcriptomics Profile in Viral Encephalitis

Step 1: Project set up with necessary packages

Step 2: Search for dataset in Gemma using individual search terms, including:

- Relevant search terms
- Specific virus names with capabilities for causing encephalitis

Datasets included for meta-analysis: GSE30577, GSE42264, GSE44331, GSE51365, GSE53784, GSE91074

Step 3: Assess gene expression range

Some datasets, especially Agilent microarray datasets, were normalized incorrectly, which may require re-normalization.

Step 4: Extract statistical contrasts of differential analysis in the datasets

These statistical contrasts allow us to check if the differential analyses are:

- Subsetted by brain region
- Included unwanted subjects
- Set up in an appropriate manner with the reference group

Based on the statistical contrasts, we might need to re-run the differential analysis

Step 5: Download the differential expression (DE) results from each dataset

Step 6: Filter DE results for rows with good gene annotation

Step 7: Extract DE results for the contrasts of interest

Step 8: Collapse DE results to one result per gene & Calculate standard error and sampling variance

Gene expression can be measured using multiple probes (microarray). Therefore, DE results need to be collapsed to one result per gene.

Standard error of $\text{Log2FC} = \text{Log2FC}/\text{T-statistics}$

Sampling variance = Average of standard error of each gene²

Step 9: Align DE results from same models (either from mouse model or rat model)

As DE results from different datasets are in slightly different orders due to various experimental factors, we want to align these results so that the DE results from each dataset are columns, with each row representing a different gene.

Step 10: Align DE results from different models (combine DE results from both mouse model and rat model using the ortholog database from Jackson Lab)

Step 11: Compare Log2FC across datasets with correlation matrix and hierarchically clustered heatmap

1) Project Set Up

```
# Load packages
library("gemma.R")
library("plyr")
library("tidyverse")
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.0      v tibble    3.2.1
## v lubridate  1.9.3      v tidyr     1.3.1
## v purrr      1.0.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::arrange() masks plyr::arrange()
## x purrr::compact() masks plyr::compact()
## x dplyr::count() masks plyr::count()
## x dplyr::desc() masks plyr::desc()
## x dplyr::failwith() masks plyr::failwith()
## x dplyr::filter() masks stats::filter()
## x dplyr::id() masks plyr::id()
## x dplyr::lag() masks stats::lag()
## x dplyr::mutate() masks plyr::mutate()
## x dplyr::rename() masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library("writexl")
library("pheatmap")
library("RColorBrewer")
library("viridis")
```

```
## Loading required package: viridisLite
```

```
# Check loaded packages
(.packages())
```

```
## [1] "viridis"      "viridisLite"  "RColorBrewer" "pheatmap"    "writexl"
## [6] "lubridate"    "forcats"      "stringr"       "dplyr"        "purrr"
## [11] "readr"        "tidyr"        "tibble"        "ggplot2"      "tidyverse"
## [16] "plyr"         "gemma.R"      "stats"         "graphics"     "grDevices"
## [21] "utils"        "datasets"     "methods"       "base"
```

2) Gemma Dataset Search

```
# Extract datasets from Gemma based on single search terms
Sepsis <- searchDatasets("Sepsis", limit = 100)
Encephalitis <- searchDatasets("Encephalitis", limit = 100)
Viral_Encephalitis <- searchDatasets("Viral Encephalitis", limit = 100)
Neuroinflammation <- searchDatasets("Neuroinflammation", limit = 100)
Neuroimmune <- searchDatasets("Neuroimmune", limit = 100)
Neuroimmunology <- searchDatasets("Neuroimmunology", limit = 100)
Neurotoxicity <- searchDatasets("Neurotoxicity", limit = 100)
Neuroimmunity <- searchDatasets("Neuroimmunity", limit = 100)
Neuroinflamm <- searchDatasets("Neuroinflamm", limit = 100)
Neuroinflammatory <- searchDatasets("Neuroinflammatory", limit = 100)
Neurotropic <- searchDatasets("Neurotropic", limit = 100)
Neuroinvasive <- searchDatasets("Neuroinvasive", limit = 100)
Neurovirulence <- searchDatasets("Neurovirulence", limit = 100)
Viral <- searchDatasets("Viral", limit = 100)
Virus <- searchDatasets("Virus", limit = 100)
Infection <- searchDatasets("Infection", limit = 100)
Viral_Infection <- searchDatasets("Viral Infection", limit = 100)
Brain_Infection <- searchDatasets("Brain Infection", limit = 100)
Brain_Inflammation <- searchDatasets("Brain Inflammation", limit = 100)
Brain_Immunology <- searchDatasets("Brain Immunology", limit = 100)
Brain_Inflammatory <- searchDatasets("Brain Inflammatory", limit = 100)

# Extract datasets from Gemma based on specific viruses
Zika <- searchDatasets("Zika", limit = 100)
Chikungunya <- searchDatasets("Chikungunya", limit = 100)
West_Nile <- searchDatasets("West Nile", limit = 100)
Japanese_encephalitis <- searchDatasets("Japanese encephalitis", limit = 100)
Venezuelan_encephalitis <- searchDatasets("Venezuelan encephalitis", limit = 100)
StLouis_encephalitis <- searchDatasets("St Louis encephalitis", limit = 100)
Polio <- searchDatasets("Polio", limit = 100)
Mumps <- searchDatasets("Mumps", limit = 100)
Measles <- searchDatasets("Measles", limit = 100)
Nipah <- searchDatasets("Nipah", limit = 100)
Hendra <- searchDatasets("Hendra", limit = 100)
Herpes <- searchDatasets("Herpes", limit = 100)
Varicella <- searchDatasets("Varicella zoster", limit = 100)
Epstein_Barr <- searchDatasets("Epstein Barr", limit = 100)
Cytomegalo <- searchDatasets("Cytomegalovirus", limit = 100)
Rabies <- searchDatasets("Rabies", limit = 100)
Influenza <- searchDatasets("Influenza", limit = 100)
Corona <- searchDatasets("Coronavirus", limit = 100)
Yellow_fever <- searchDatasets("Yellow fever", limit = 100)

# Extract the data to excel file
write_xlsx(Sepsis, "Sepsis.xlsx")
write_xlsx(Encephalitis, "Encephalitis.xlsx")
write_xlsx(Viral_Encephalitis, "Viral Encephalitis.xlsx")
write_xlsx(Neuroinflammation, "Neuroinflammation.xlsx")
```

```

write_xlsx(Neuroimmunity, "Neuroimmunity.xlsx")
write_xlsx(Neuroinflammatory, "Neuroinflammatory.xlsx")
write_xlsx(Neurotoxicity, "Neurotoxicity.xlsx")
write_xlsx(Neurotropic, "Neurotropic.xlsx")
write_xlsx(Neuroinvasive, "Neuroinvasive.xlsx")
write_xlsx(Neurovirulence, "Neurovirulence")
write_xlsx(Viral, "Viral.xlsx")
write_xlsx(Viral_Infection, "Viral_Infection.xlsx")
write_xlsx(Virus, "Virus.xlsx")
write_xlsx(Infection, "Infection.xlsx")

# Count the frequency of unique values in a dataframe
table(Sepsis$taxon.Name) #Count based on the model (human, mouse, rat)

# Combine and exclude datasets of other models than mouse/rat
Total_datasets = rbind(Brain_Immunology, Brain_Infection, Brain_Inflammation,
                        Corona, Cytomegalo, Encephalitis, Epstein_Barr, Hendra,
                        Herpes, Infection, Influenza, Japanese_encephalitis, Measles,
                        Neuroinflammation, Neuroinflammatory, Neuroinvasive,
                        Neurotoxicity, Neurotropic, Neurovirulence, Rabies, Sepsis,
                        Varicella, Venezuelan_encephalitis, Viral, Viral_Encephalitis,
                        Viral_Infection, Virus, West_Nile, Yellow_fever, Zika)
Total_datasets_eli_1 = subset(Total_datasets, Total_datasets$taxon.Name == "mouse")
Total_datasets_eli_2 = subset(Total_datasets, Total_datasets$taxon.Name == "rat")
Total_datasets_eli = unique(rbind(Total_datasets_eli_1, Total_datasets_eli_2))
write_xlsx(Total_datasets_eli, "Total_datasets.xlsx")

```

Through process of inclusion/exclusion criteria, 6 datasets are included in the meta-analysis, including: GSE30577, GSE42264, GSE44331, GSE51365, GSE53784, GSE91074.

3) Gene Expression Range Assessment

```

# Create a function to check the expression range of the dataset
# (especially for the Agilent microarray datasets)
CheckGeneExpressionRange <- function(dataset_shortcode){

  # Retrieve the processed expression data for the given dataset
  expression_data <- get_dataset_processed_expression(dataset_shortcode)

  # Print the structure of the expression data
  print(str(expression_data))

  # The first four columns are row metadata: Probe, GeneSymbol, GeneName, NCBI ID
  # The rest of the columns are gene expression values for each subject

  # Exclude metadata row (row 1-4) and convert the gene expression columns to a
  # matrix for further analysis
  expression_matrix <- as.matrix(expression_data[, -1:-4])

  # Create a histogram of the expression data
  hist(expression_matrix,

```

```

main = paste("Histogram of Expression Data for", dataset_shortname),
# The y-axis is the gene frequency
# The x-axis is log 2 gene expression - log 2 counts per million
xlab = "Log 2 Expression", ylab = "Frequency")

# The large spike on the left side of the histogram ("floor effect") are
# all of the genes that are not truly expressed or have too low of expression
# to be measurable

# Log 2 RNA-seq dataset has the range between -5 to 12
# Log 2 Microarray dataset has the range between 4 to 15

# Calculate the min, median, max values of the expression data
min_val <- min(expression_matrix)
median_val <- median(expression_matrix)
max_val <- max(expression_matrix)

# Print the calculated values
print(paste("Minimum value:", min_val))
print(paste("Median value:", median_val))
print(paste("Maximum value:", max_val))
}

```

```

# Function use
CheckGeneExpressionRange("GSE30577") # Min: -4.8 / Max: 8.6 / Agilent Microarray

```

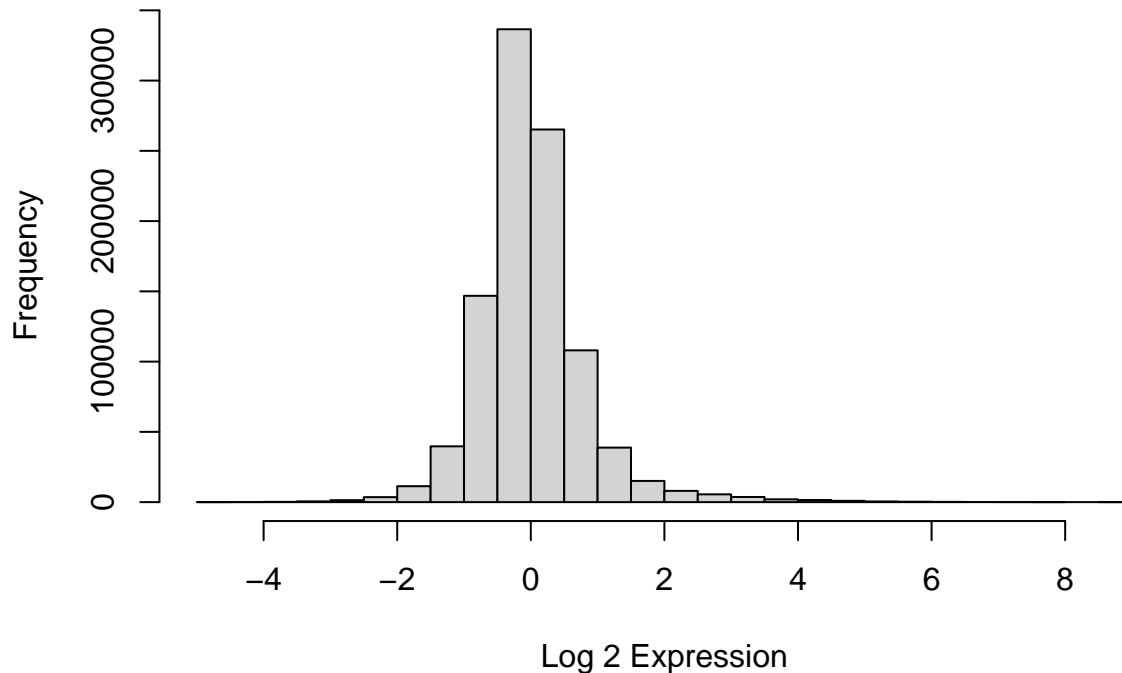
```

## Classes 'data.table' and 'data.frame':  41252 obs. of  28 variables:
## $ Probe                : chr  "AT_D_3" "AT_D_5" "AT_D_M" "AT_L_3" ...
## $ GeneSymbol            : chr  "" "" "" "" ...
## $ GeneName              : chr  "" "" "" "" ...
## $ NCBIId                : chr  "" "" "" "" ...
## $ Rabies, Fix, Spinalcord, 7dpi(mock), 06: num  0.175 0.732 0.34 0.219 0.893 ...
## $ Rabies, Fix, Spinalcord, 7dpi(mock), 05: num -0.4457 0.0977 0.4108 0.0203 0.8468 ...
## $ Rabies, Fix, Spinalcord, 7dpi(mock), 04: num -0.845 -0.335 -0.441 0.251 -0.375 ...
## $ Rabies, Fix, Spinalcord, 7dpi(mock), 03: num  0.554 -0.353 -0.634 -0.243 -0.774 ...
## $ Rabies, Fix, Spinalcord, 7dpi(mock), 02: num  0.822 -0.283 0.209 -0.627 -0.54 ...
## $ Rabies, Fix, Spinalcord, 7dpi(mock), 01: num -0.567 0.022 0.049 -0.0055 0.0596 ...
## $ Rabies, Fix, Spinalcord, 7dpi(inf), 06 : num -0.4334 -0.0176 -0.0265 -0.1352 -0.0863 ...
## $ Rabies, Fix, Spinalcord, 7dpi(inf), 05 : num  0.541 0.809 0.94 0.77 0.838 ...
## $ Rabies, Fix, Spinalcord, 7dpi(inf), 04 : num  0.7378 -0.067 -0.0598 -0.1549 0.0719 ...
## $ Rabies, Fix, Spinalcord, 7dpi(inf), 03 : num  0.0658 0.3534 0.4016 0.7704 0.3902 ...
## $ Rabies, Fix, Spinalcord, 7dpi(inf), 02 : num  0.6398 -0.1694 -0.218 -0.0018 -0.3153 ...
## $ Rabies, Fix, Spinalcord, 7dpi(inf), 01 : num -0.6072 -0.111 -0.2676 -0.3819 -0.0674 ...
## $ Rabies, Fix, Brain, 7dpi(mock), 06      : num -0.507 -0.428 -0.427 -0.542 -0.416 ...
## $ Rabies, Fix, Brain, 7dpi(mock), 05      : num -0.85 -0.644 -0.637 -0.707 -0.68 ...
## $ Rabies, Fix, Brain, 7dpi(mock), 04      : num  0.809 0.346 0.36 0.247 1.073 ...
## $ Rabies, Fix, Brain, 7dpi(mock), 03      : num -0.149 -0.412 -0.316 -0.586 -0.552 ...
## $ Rabies, Fix, Brain, 7dpi(mock), 02      : num  0.602 0.443 0.396 0.47 0.322 ...
## $ Rabies, Fix, Brain, 7dpi(mock), 01      : num  0.195 0.761 0.764 1.686 0.698 ...
## $ Rabies, Fix, Brain, 7dpi(inf), 06      : num  0.315 0.491 0.44 0.297 0.368 ...
## $ Rabies, Fix, Brain, 7dpi(inf), 05      : num -0.262 -0.332 -0.137 -0.522 -0.49 ...
## $ Rabies, Fix, Brain, 7dpi(inf), 04      : num  0.9652 -0.1483 -0.7497 0.0347 -0.8159 ...
## $ Rabies, Fix, Brain, 7dpi(inf), 03      : num -0.0367 -0.0308 -0.1867 0.0664 -0.221 ...

```

```
## $ Rabies, Fix, Brain, 7dpi(inf), 02      : num  -0.2082 0.0844 0.1173 0.0366 0.0158 ...
## $ Rabies, Fix, Brain, 7dpi(inf), 01      : num   0.7341 0.3499 0.2119 -0.0955 -0.0283 ...
## - attr(*, ".internal.selfref")=<externalptr>
## - attr(*, "call")= chr "https://gemma.msl.ubc.ca/rest/v2/datasets/GSE30577/data/processed"
## - attr(*, "env")=<environment: 0x10d279b20>
## NULL
```

Histogram of Expression Data for GSE30577



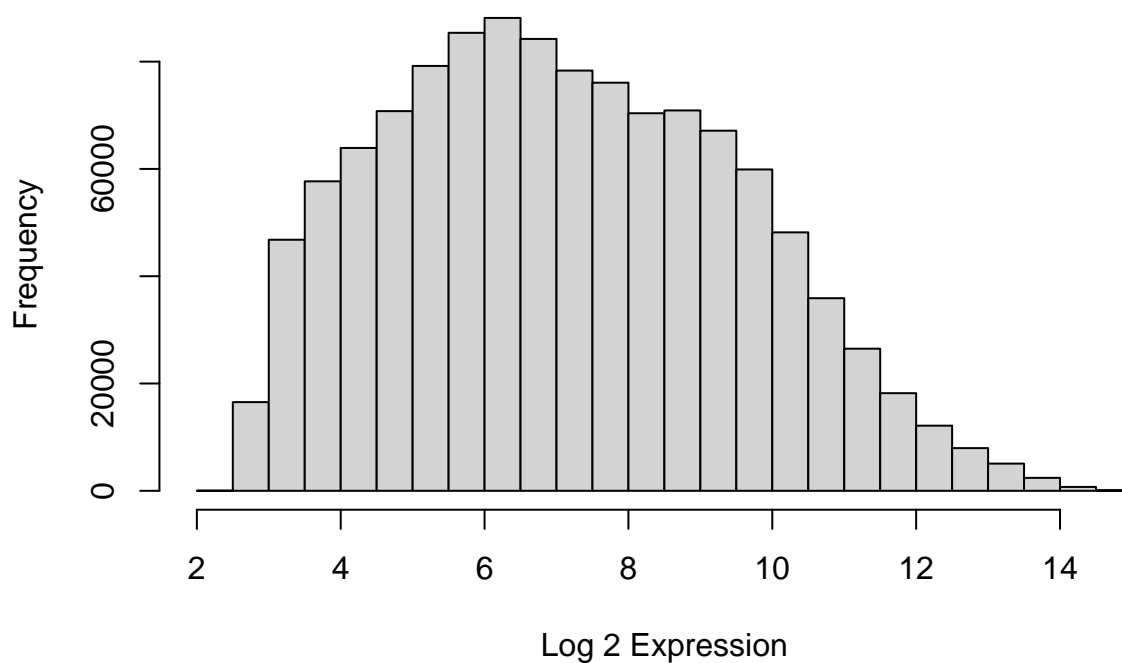
```
## [1] "Minimum value: -4.8043"
## [1] "Median value: -0.0612"
## [1] "Maximum value: 8.6806"
```

```
CheckGeneExpressionRange("GSE42264") # Min: 2.4 / Max: 14.6 / Affymetrix Array
```

```
## Classes 'data.table' and 'data.frame':  45101 obs. of  30 variables:
## $ Probe                : chr  "1415670_at" "1415671_at" "1415672_at" "1415673_at" ...
## $ GeneSymbol            : chr  "Copg1" "Atp6v0d1" "Golga7" "Psph" ...
## $ GeneName              : chr  "coatamer protein complex, subunit gamma 1" "ATPase, H+ tran
## $ NCBIId                : chr  "54161" "11972" "57437" "100678" ...
## $ Uninfected NT at 5 d.p.i., rep4: num  10.16 12.24 11.79 9.04 10.77 ...
## $ Uninfected NT at 5 d.p.i., rep3: num  10.25 12.2 11.69 9.03 10.67 ...
## $ Uninfected NT at 5 d.p.i., rep2: num  10.13 12.22 11.79 9.04 10.78 ...
## $ Uninfected NT at 5 d.p.i., rep1: num  10.14 12.21 11.86 9.06 10.85 ...
## $ Uninfected TG at 5 d.p.i., rep4: num  10.09 12.31 11.87 9.12 10.6 ...
## $ Uninfected TG at 5 d.p.i., rep3: num  10.23 12.26 11.78 9.04 10.59 ...
## $ Uninfected TG at 5 d.p.i., rep2: num  10.23 12.37 11.85 9.04 10.7 ...
```

```
## $ Uninfected TG at 5 d.p.i., rep1: num 10.12 12.33 11.9 9.06 10.64 ...
## $ Infected NT 10 d.p.i., rep9 : num 10.22 12.28 11.96 9.08 10.53 ...
## $ Infected NT 10 d.p.i., rep8 : num 10.28 12.3 12.05 9.08 10.54 ...
## $ Infected NT 10 d.p.i., rep7 : num 10.3 12.26 11.94 9.18 10.43 ...
## $ Infected NT 10 d.p.i., rep6 : num 10.2 12.3 12.1 9.1 10.6 ...
## $ Infected NT 10 d.p.i., rep5 : num 10.21 12.24 11.99 9.09 10.48 ...
## $ Infected NT 5 d.p.i., rep4 : num 10.21 12.18 11.81 9.08 10.7 ...
## $ Infected NT 5 d.p.i., rep3 : num 10.23 12.17 11.8 9.01 10.74 ...
## $ Infected NT 5 d.p.i., rep2 : num 10.17 12.13 11.88 9.01 10.78 ...
## $ Infected NT 5 d.p.i., rep1 : num 10.14 12.21 11.89 9.08 10.81 ...
## $ Infected TG 10 d.p.i., rep9 : num 10.24 12.29 12.07 9.07 10.47 ...
## $ Infected TG 10 d.p.i., rep8 : num 10.25 12.3 12.12 8.99 10.42 ...
## $ Infected TG 10 d.p.i., rep7 : num 10.3 12.3 12 9 10.6 ...
## $ Infected TG 10 d.p.i., rep6 : num 10.24 12.26 12.04 8.98 10.45 ...
## $ Infected TG 10 d.p.i., rep5 : num 10.28 12.33 12.04 9.04 10.49 ...
## $ Infected TG 5 d.p.i., rep4 : num 10.24 12.15 11.75 8.94 10.78 ...
## $ Infected TG 5 d.p.i., rep3 : num 10.11 12.2 11.89 9.02 10.83 ...
## $ Infected TG 5 d.p.i., rep2 : num 10.08 12.26 11.89 9.07 10.9 ...
## $ Infected TG 5 d.p.i., rep1 : num 10.16 12.22 11.87 9.06 10.84 ...
## - attr(*, ".internal.selfref")=<externalptr>
## - attr(*, "call")= chr "https://gemma.msl.ubc.ca/rest/v2/datasets/GSE42264/data/processed"
## - attr(*, "env")=<environment: 0x12c867000>
## NULL
```

Histogram of Expression Data for GSE42264



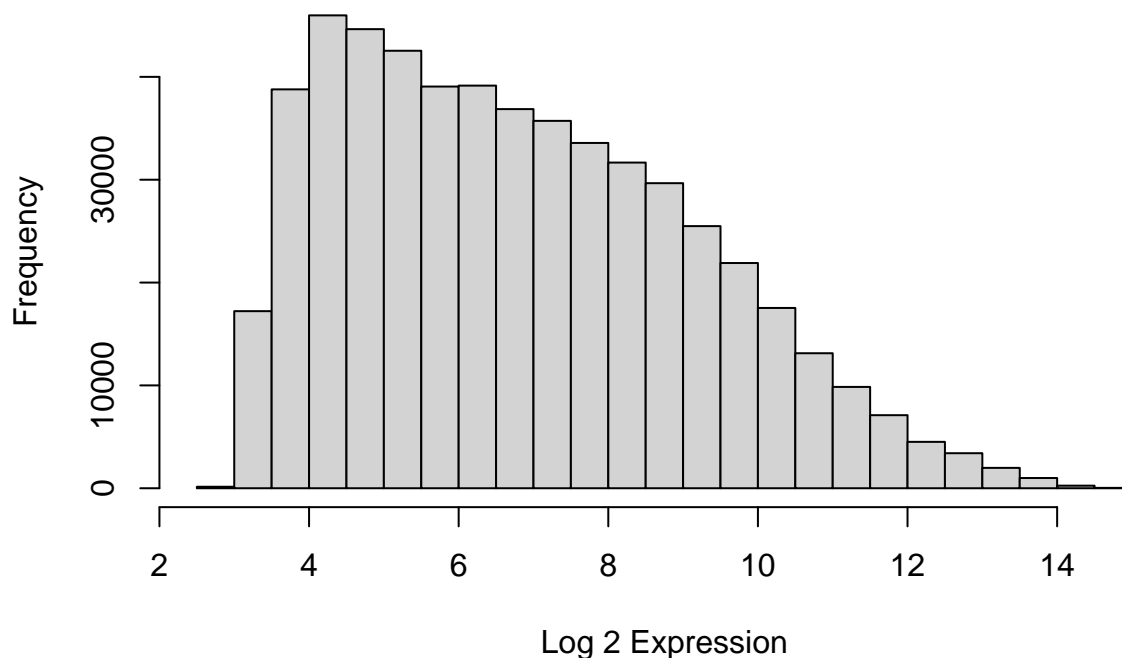
```
## [1] "Minimum value: 2.4635"
## [1] "Median value: 6.9608"
```

```
## [1] "Maximum value: 14.685"
```

```
CheckGeneExpressionRange("GSE44331") # Min: 2.9 / Max: 14.5 / Affymetrix Array
```

```
## Classes 'data.table' and 'data.frame': 45101 obs. of 16 variables:
## $ Probe : chr "1415670_at" "1415671_at" "1415672_at" "1415673_at"
## $ GeneSymbol : chr "Copg1" "Atp6v0d1" "Golga7" "PspH" ...
## $ GeneName : chr "coatamer protein complex, subunit gamma 1" "ATPase
## $ NCBIid : chr "54161" "11972" "57437" "100678" ...
## $ KO VSV infected, biological replicate 3: num 9.1 11.98 11.74 8.74 10.1 ...
## $ KO VSV infected, biological replicate 2: num 9.14 12.07 11.76 8.69 9.99 ...
## $ KO VSV infected, biological replicate 1: num 9.06 12 11.88 8.82 9.99 ...
## $ WT VSV infected, biological replicate 3: num 9.09 12.03 11.72 8.8 10.02 ...
## $ WT VSV infected, biological replicate 2: num 9 12.1 11.75 8.77 10.07 ...
## $ WT VSV infected, biological replicate 1: num 9.08 12.06 11.79 8.74 10.09 ...
## $ KO uninfected, biological replicate 3 : num 9.13 12.07 11.84 8.77 10.04 ...
## $ KO uninfected, biological replicate 2 : num 9.13 12.03 11.89 8.8 10.01 ...
## $ KO uninfected, biological replicate 1 : num 9.19 11.98 11.94 8.77 10.03 ...
## $ WT uninfected, biological replicate 3 : num 9.07 12.07 11.79 8.74 9.99 ...
## $ WT uninfected, biological replicate 2 : num 9.12 12.06 11.87 8.83 10.22 ...
## $ WT uninfected, biological replicate 1 : num 9.09 12.09 11.86 8.7 9.79 ...
## - attr(*, ".internal.selfref")=<externalptr>
## - attr(*, "call")= chr "https://gemma.msl.ubc.ca/rest/v2/datasets/GSE44331/data/processed"
## - attr(*, "env")=<environment: 0x12c851e00>
## NULL
```

Histogram of Expression Data for GSE44331

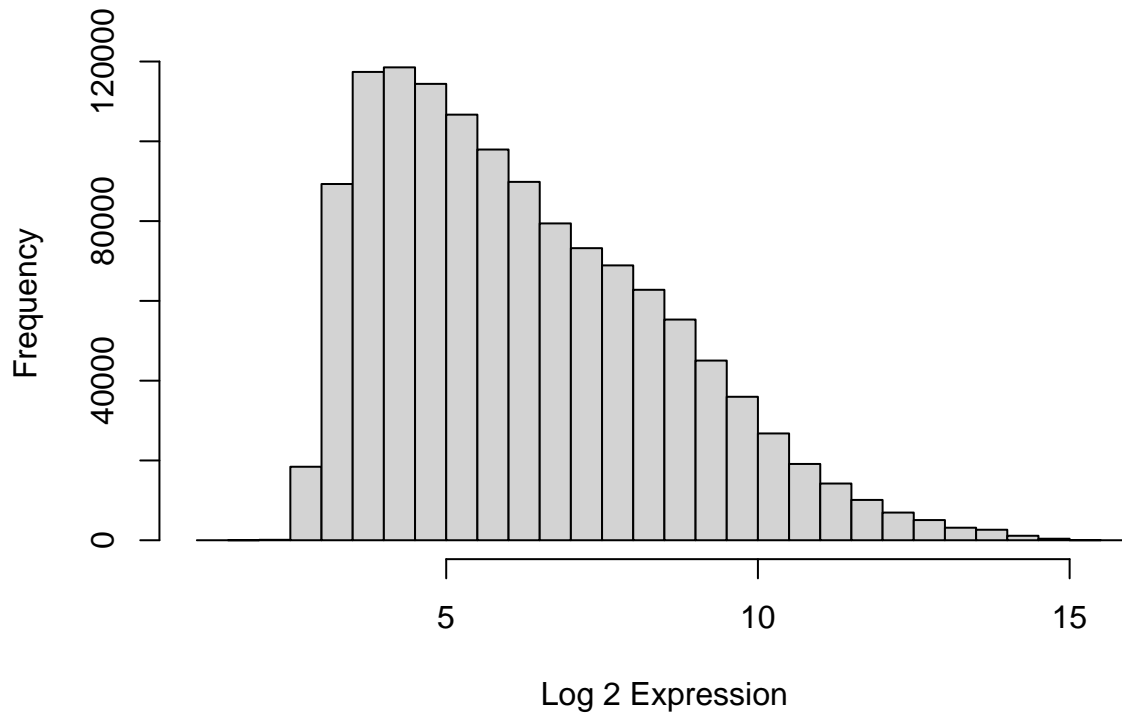



```
## [1] "Minimum value: 2.9042"
## [1] "Median value: 6.5411"
## [1] "Maximum value: 14.5935"
```

```
CheckGeneExpressionRange("GSE51365") # Min: 1.2 / Max: 15.7 / Affymetrix Array
```

```
## Classes 'data.table' and 'data.frame': 45101 obs. of 32 variables:
## $ Probe : chr "1415670_at" "1415671_at" "1415672_at" "1415673_at" ...
## $ GeneSymbol : chr "Copg1" "Atp6v0d1" "Golga7" "Psph" ...
## $ GeneName : chr "coatomer protein complex, subunit gamma 1" "ATPase, H+ transporting, 1"
## $ NCBIid : chr "54161" "11972" "57437" "100678" ...
## $ BRAIN|ORF73|SC019 : num 8.97 11.98 11.31 7.92 9.46 ...
## $ BRAIN|ORF73|SC018 : num 8.78 11.95 11.33 7.97 9.39 ...
## $ BRAIN|ORF73|SC017 : num 8.86 11.96 11.36 8.11 9.5 ...
## $ BRAIN|MHV-68|SC019 : num 8.85 11.82 11.36 8.03 9.43 ...
## $ BRAIN|MHV-68|SC018 : num 8.98 12.05 11.43 8.17 9.44 ...
## $ BRAIN|MHV-68|SC017 : num 8.89 11.99 11.37 7.99 9.21 ...
## $ BRAIN|MOCK|SC019 : num 8.7 12.16 11.32 7.88 9.65 ...
## $ BRAIN|MOCK|SC018 : num 8.94 11.86 11.46 8.09 9.4 ...
## $ BRAIN|MOCK|SC017 : num 9.04 12.07 11.4 8 9.29 ...
## $ SPLEEN|ORF73|SC019 : num 8.88 10.75 11.37 7.66 9.21 ...
## $ SPLEEN|ORF73|SC3 : num 8.7 10.54 11.15 7.51 9.17 ...
## $ SPLEEN|ORF73|SC2 : num 8.62 10.45 11.08 7.53 8.99 ...
## $ SPLEEN|MHV-68|SC019 : num 8.97 10.62 11.22 7.82 9.18 ...
## $ SPLEEN|MHV-68|SC3 : num 9.12 10.75 11.34 7.78 9.17 ...
## $ SPLEEN|MHV-68|SC2 : num 8.87 10.62 11.05 7.67 8.92 ...
## $ SPLEEN|MOCK|SC019 : num 8.86 10.61 11.27 7.85 9.33 ...
## $ SPLEEN|MOCK|SC3 : num 8.72 10.61 11.28 7.48 9.12 ...
## $ SPLEEN|MOCK|SC2 : num 8.58 10.51 11 7.52 9.11 ...
## $ LIVER|ORF73|SC019 : num 9.26 10.61 10.89 7.33 9.22 ...
## $ LIVER|ORF73|SC3 : num 9.61 10.75 11.03 7.29 9.27 ...
## $ LIVER|ORF73|SC2 : num 9.41 10.88 11.05 7.41 9.33 ...
## $ LIVER|MHV-68|SC019 : num 9.31 10.62 10.91 7 9.02 ...
## $ LIVER|MHV-68|SC2 : num 9.49 10.66 11.06 7.2 9.22 ...
## $ LIVER|MHV-68|SC3 : num 9.43 10.65 11.09 7 9.08 ...
## $ LIVER|MOCK|SC019 : num 9.48 10.66 10.89 7.07 9.06 ...
## $ LIVER|MOCK|SC3 : num 9.54 10.85 10.89 7.42 9.3 ...
## $ LIVER|MOCK|SC2 : num 9.61 10.73 10.91 7.43 9.34 ...
## $ LIVER|MOCK|SC1 : num 9.43 10.75 11.01 7.21 9.17 ...
## - attr(*, ".internal.selfref")=<externalptr>
## - attr(*, "call")= chr "https://gemma.msl.ubc.ca/rest/v2/datasets/GSE51365/data/processed"
## - attr(*, "env")=<environment: 0x10e312640>
## NULL
```

Histogram of Expression Data for GSE51365

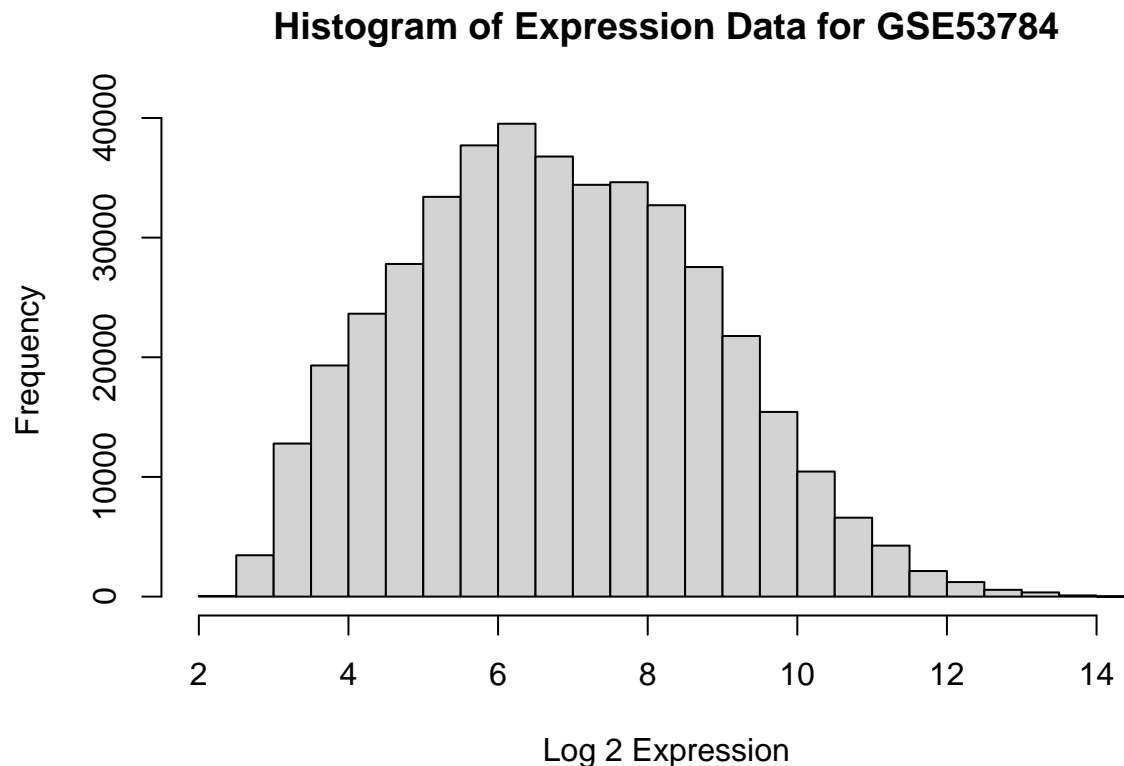


```
## [1] "Minimum value: 1.2468"
## [1] "Median value: 5.8355"
## [1] "Maximum value: 15.7641"
```

```
CheckGeneExpressionRange("GSE53784") # Min: 2.0 / Max: 14.1 / Affymetrix Array
```

```
## Classes 'data.table' and 'data.frame': 35556 obs. of 16 variables:
## $ Probe      : int 10338001 10338002 10338003 10338004 10338005 10338006 10338007 10338008 10338009
## $ GeneSymbol: chr "" "" "" "" ...
## $ GeneName   : chr "" "" "" "" ...
## $ NCBIid     : chr "" "" "" "" ...
## $ WNV_B8     : num 11.61 6.9 9.88 8.63 2.98 ...
## $ WNV_B7     : num 12 6.85 10.31 9.23 3.02 ...
## $ WNV_B6     : num 11.91 7.08 10.32 9.14 3 ...
## $ Mock_B6    : num 11.82 7.14 10.09 9.2 2.89 ...
## $ Mock_B5    : num 12.01 7.15 10.53 9.55 2.97 ...
## $ Mock_B4    : num 11.86 7 10.22 9.08 2.93 ...
## $ MOCK_6     : num 12.07 6.98 10.49 9.49 2.92 ...
## $ MOCK_5     : num 11.73 6.95 10.03 9.11 2.93 ...
## $ MOCK_4     : num 11.71 7.26 10.12 9.07 2.93 ...
## $ JEV_3      : num 11.77 7.3 10.18 8.79 2.93 ...
## $ JEV_2      : num 11.92 7.23 10.24 9.16 2.97 ...
## $ JEV_1      : num 11.9 7.41 10.4 9.05 2.93 ...
## - attr(*, ".internal.selfref")=<externalptr>
## - attr(*, "call")= chr "https://gemma.msl.ubc.ca/rest/v2/datasets/GSE53784/data/processed"
```

```
## - attr(*, "env")=<environment: 0x12bdb3b20>
## NULL
```



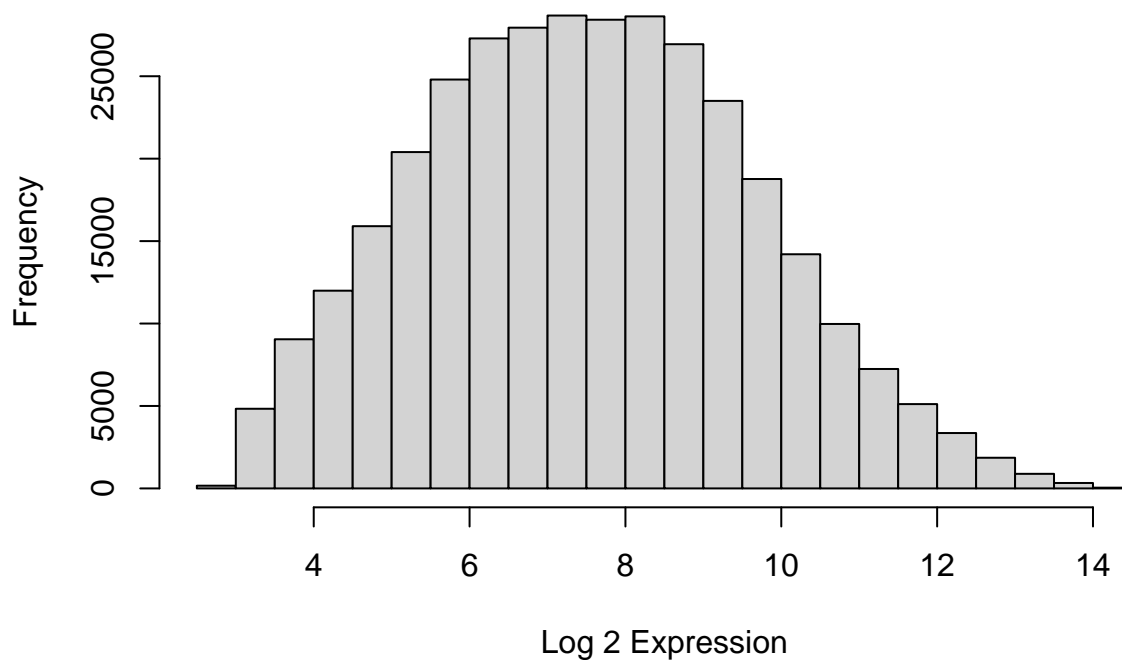
```
## [1] "Minimum value: 2.0069"
## [1] "Median value: 6.7054"
## [1] "Maximum value: 14.1214"
```

```
CheckGeneExpressionRange("GSE91074") # Min: 2.8 | Max: 14.3 | Affymetrix Array
```

```
## Classes 'data.table' and 'data.frame': 22690 obs. of 19 variables:
## $ Probe : chr "1415670_at" "1415671_at" "1415672_at" ...
## $ GeneSymbol : chr "Copg1" "Atp6v0d1" "Golga7" "Psph" ...
## $ GeneName : chr "coatamer protein complex, subunit gamma" ...
## $ NCBIId : chr "54161" "11972" "57437" "100678" ...
## $ C57BL6/J mouse mock)-infected, biological rep2 : num 9.41 11.35 11.55 8 9.25 ...
## $ C57BL6/J mouse mock)-infected, biological rep1 : num 9.43 11.46 11.65 8.07 9.11 ...
## $ C57BL6/J mouse VEEV (TC83)-infected, biological rep3 : num 9.34 11.09 11.3 7.9 9.04 ...
## $ C57BL6/J mouse VEEV (TC83)-infected, biological rep2 : num 9.34 11.06 11.45 7.9 9.06 ...
## $ C57BL6/J mouse VEEV (TC83)-infected, biological rep1 : num 9.27 11 11.34 8.04 9.18 ...
## $ ??-TCR -/- mouse mock-infected, biological rep2 : num 9.21 11.18 11.54 8.07 9.3 ...
## $ ??-TCR -/- mouse mock-infected, biological rep1 : num 9.12 11.3 11.48 8.17 9.08 ...
## $ ??-TCR -/- mouse VEEV (TC83)-infected, biological rep3 : num 8.96 10.96 11.44 7.89 8.94 ...
## $ ??-TCR -/- mouse VEEV (TC83)-infected, biological rep2 : num 9.44 11.11 11.57 7.74 9.17 ...
## $ ??-TCR -/- mouse VEEV (TC83)-infected, biological rep1 : num 9.2 11.18 11.27 7.92 9.05 ...
## $ C3H/HeN mouse mock-infected, biological rep2 : num 9.2 11.32 11.48 8.07 9.05 ...
```

```
## $ C3H/HeN mouse mock-infected, biological rep1 : num 9.32 11.45 11.63 8.14 9.04 ...
## $ C3H/HeN mouse VEEV (TC83)-infected, biological rep3 : num 9.54 11.22 11.48 7.85 9.27 ...
## $ C3H/HeN mouse VEEV (TC83)-infected, biological rep2 : num 9.28 11.13 11.43 7.72 9.08 ...
## $ C3H/HeN mouse VEEV (TC83)-infected, biological rep1 : num 9.15 11.16 11.48 7.79 9.01 ...
## - attr(*, ".internal.selfref")=<externalptr>
## - attr(*, "call")= chr "https://gemma.msl.ubc.ca/rest/v2/datasets/GSE91074/data/processed"
## - attr(*, "env")=<environment: 0x12bdc8028>
## NULL
```

Histogram of Expression Data for GSE91074



```
## [1] "Minimum value: 2.8577"
## [1] "Median value: 7.4844"
## [1] "Maximum value: 14.3048"
```

Gene expression of all 6 datasets seem to be within normal range and have appropriate distribution.

4) Extraction of Statistical Contrasts of Differential Analysis in Each Dataset

For the meta-analysis, we will be extracting the differential expression results from Gemma. The differential expression results for each dataset may include multiple result sets (e.g., one result set for the subset of the data from the hippocampus, one result set for frontal cortex). Each of these result sets may have multiple statistical contrasts (e.g., drug1 vs. vehicle, drug2 vs. vehicle). Therefore, each of the statistical contrasts is labeled with a result ID and contrast ID within the Gemma database. We will need to know which of these IDs are relevant to our project goals to easily extract their results.

We will also need to double-check that these statistical contrasts are set up in a manner that makes sense for our experiments:

1. For experiments that include more than one brain region, we will need to double-check that the results have been subsetted by brain region (instead of including brain region ("OrganismPart") as a factor in the model). If they have not been subsetted by region, we will probably need to re-run the differential expression analysis.
2. Depending on the goals of the meta-analysis, we may also need to re-run the differential expression analysis to remove other unwanted subjects (e.g., removing subjects with genotypes that might interfere with our results).
3. We will need to double-check that the comparisons include an appropriate reference group - sometimes they are reversed in Gemma (e.g., having the drug treatment set as the baseline, with vehicle as the manipulation). If this is the case, we will need to invert the effects when we input them into our meta-analysis (multiply the effects by -1).

```
# Create a function to extract statistical contrasts of differential analysis
# in each dataset
GettingResultSetInfoForDatasets <- function(ExperimentIDs) {

  # Create a dataframe to store the results
  resultsets_toscreen <- data.frame(
    ExperimentIDs = character(),
    ResultSetIDs = character(),
    ContrastIDs = character(),
    FactorCategory = character(),
    ExperimentalFactors = character(),
    BaselineFactors = character(),
    Subsetted = logical(),
    SubsetBy = character(),
    stringsAsFactors = FALSE
  )

  # Loop through each of the datasets
  for (i in c(1:length(ExperimentIDs))) {

    # Extract the differential expression analysis from the datasets
    design <- get_dataset_differential_expression_analyses(ExperimentIDs[i])

    # Check if the design dataframe has rows
    if (nrow(design) > 0) {

      # Create the vectors to store the experimental and baseline factors
      experimental_factors <- vector(mode = "character", length = nrow(design))
      baseline_factors <- vector(mode = "character", length = nrow(design))

      # Loop through each result.ID in the design dataframe
      for (j in c(1:nrow(design))) {

        # Concatenate the experimental factors into a single string
        experimental_factors[j] <- paste(design$experimental.factors[[j]]$summary,
                                          collapse = ";")

        # Concatenate the baseline factors into a single string
        baseline_factors[j] <- paste(design$baseline.factors[[j]]$summary,
                                     collapse = ";")
      }
    }
  }
}
```

```

# Create a vector to store subset information
SubsetBy <- vector(mode = "character", length = nrow(design))

# Check if the design dataframe is subsetting or not
if (design$isSubset[1] == TRUE) {

  # Loop through each result.ID to extract and concatenate subset information
  for (j in c(1:nrow(design))) {
    SubsetBy[j] <- paste(design$subsetFactor[[j]]$summary, collapse = ";")
  }
} else {
  # If not subsetting, fill the SubsetBy vector with NA values
  SubsetBy <- rep(NA, length(design$result.ID))
}

# Create a temporary dataframe to store extracted info
resultsets_for_experiment <- data.frame(
  ExperimentIDs = ExperimentIDs[i],
  ResultSetIDs = design$result.ID,
  ContrastIDs = design$contrast.ID,
  FactorCategory = design$factor.category,
  ExperimentalFactors = experimental_factors,
  BaselineFactors = baseline_factors,
  Subsetting = design$isSubset,
  SubsetBy = SubsetBy,
  stringsAsFactors = FALSE
)

# Append the temporary dataframe with the dataframe created initially
resultsets_toscreen <- rbind(resultsets_toscreen, resultsets_for_experiment)
}

}

# Add empty columns to store screening notes
resultsets_toscreen <- cbind(
  resultsets_toscreen,
  Include = vector(mode = "character", length = nrow(resultsets_toscreen)),
  WrongBaseline = vector(mode = "character", length = nrow(resultsets_toscreen)),
  ResultsNotRegionSpecific = vector(mode = "character", length = nrow(resultsets_toscreen)),
  ReAnalyze = vector(mode = "character", length = nrow(resultsets_toscreen)),
  stringsAsFactors = FALSE
)

# Export the final dataframe into the working directory
write.csv(resultsets_toscreen, "ResultSets_toScreen.csv")

# Print a message to indicate that the results have been saved
print("Output object: ResultSets_toScreen.csv")
}

# Set up a vector of the names of all datasets
ExperimentIDs <- c("GSE30577", "GSE42264", "GSE44331",

```

```

"GSE51365", "GSE53784", "GSE91074")

# Function use
GettingResultSetInfoForDatasets(ExperimentIDs)

```

```
## [1] "Output object: ResultSets_toScreen.csv"
```

5) Downloading the DE Results from Each Dataset

```

# Create a function to download DE results and extract Log2FC and T-statistics
# for contrasts of interest
DownloadingDEResults <- function(ResultSets_contrasts){

  # Identify the unique ResultSet IDs
  # Some ResultSets may have multiple statistical contrasts, but we only want unique ResultSet IDs
  UniqueResultSetIDs <- unique(ResultSets_contrasts$ResultSetIDs)

  # Print the identified unique ResultSet IDs
  print("ResultSets identified as being of interest")
  print(UniqueResultSetIDs)

  # Download DE results for each unique ResultSet ID
  differentials <- UniqueResultSetIDs %>%
    # The function returns a list because single experiment may have multiple
    # resultSets
    # Only take the first element of the output
    # The "resultSet" argument is used to directly access the results we need
    lapply(function(x) {get_differential_expression_values(resultSet = x)[[1]]})

  # Some datasets might not have all the advertised DE results due to a variety of
  # so we need to remove empty differential (only keep differential with rows present)
  non_missing_contrasts <- sapply(differentials, function(df) nrow(df) > 0)

  # Return the "differentials" object that contains the DE results of contrast of interest
  differentials <- differentials[non_missing_contrasts]
  UniqueResultSetIDs <- UniqueResultSetIDs[non_missing_contrasts]

  # Print the ResultSet IDs that had DE results
  print("ResultSets that had DE results:")
  print(UniqueResultSetIDs)

  # Print a message to inform the structure of the output "differentials"
  print("Your DE results for each of the ResultSets are stored in object differentials.")
  print("This object is structured as a list of dataframes.")
  print("Each element in the list represents a ResultSet, with the dataframe containing DE results")

  # Extract the effect sizes of Log2FC of contrasts of interest
  print("Columns of effect sizes (Log2FC) for contrasts of interest:")
  Contrasts_Log2FC <- paste("contrast_", ResultSets_contrasts$ContrastIDs,
                           "_log2fc", sep = "")
  print(Contrasts_Log2FC)

```

```

# Extract the T-statistics of contrasts of interest for calculating sampling variances
print("Columns of T-statistics for contrasts of interest:")
Contrasts_Tstat <- paste("contrast_", ResultSets_contrasts$ContrastIDs,
                        "_tstat", sep = "")
print(Contrasts_Tstat)

# Remove the temporary Log2FC and T-statistics of contrast of interest
rm(Contrasts_Log2FC, Contrasts_Tstat)
}

```

```

# Import the CSV file with contrasts of interest
ResultSets_contrasts <- read.csv("ResultSets_Screened.csv",
                                header = TRUE, stringsAsFactors = FALSE)

# Function use
DownloadingDEResults(ResultSets_contrasts)

```

```

## [1] "ResultSets identified as being of interest"
## [1] 535864 513873 546199 466082 522293 523663
## [1] "ResultSets that had DE results:"
## [1] 535864 513873 546199 466082 522293 523663
## [1] "Your DE results for each of the ResultSets are stored in object differentials."
## [1] "This object is structured as a list of dataframes."
## [1] "Each element in the list represents a ResultSet, with the dataframe containing DE results"
## [1] "Columns of effect sizes (Log2FC) for contrasts of interest:"
## [1] "contrast_84556_log2fc" "contrast_89756_log2fc" "contrast_97695_log2fc"
## [4] "contrast_99431_log2fc" "contrast_99432_log2fc" "contrast_140573_log2fc"
## [7] "contrast_140572_log2fc" "contrast_142941_log2fc"
## [1] "Columns of T-statistics for contrasts of interest:"
## [1] "contrast_84556_tstat" "contrast_89756_tstat" "contrast_97695_tstat"
## [4] "contrast_99431_tstat" "contrast_99432_tstat" "contrast_140573_tstat"
## [7] "contrast_140572_tstat" "contrast_142941_tstat"

```

```

# Check the structure of the "differentials" object
str(differentials)

```

```

## List of 6
## $ :Classes 'data.table' and 'data.frame': 22883 obs. of 11 variables:
## ..$ Probe : chr [1:22883] "A_51_P109508" "A_52_P185343" "A_52_P212756" "A_51_P
## ..$ NCBIid : chr [1:22883] "171285" "14674" "240832" "14990" ...
## ..$ GeneSymbol : chr [1:22883] "Havcr2" "Gna13" "Tor1aip2" "H2-M2" ...
## ..$ GeneName : chr [1:22883] "hepatitis A virus cellular receptor 2" "guanine nuc
## ..$ pvalue : num [1:22883] 6.67e-12 4.11e-11 2.09e-11 2.18e-10 7.72e-11 ...
## ..$ corrected_pvalue : num [1:22883] 2.28e-07 3.51e-07 3.51e-07 4.37e-07 4.37e-07 ...
## ..$ rank : num [1:22883] 2.93e-05 1.00e-04 5.86e-05 4.00e-04 1.00e-04 ...
## ..$ contrast_84556_coefficient: num [1:22883] 4.96 6.96 3.53 5.19 5.27 ...
## ..$ contrast_84556_log2fc : num [1:22883] 4.96 6.96 3.53 5.19 5.27 ...
## ..$ contrast_84556_tstat : num [1:22883] 8.91 11.63 7.13 9.77 9.96 ...
## ..$ contrast_84556_pvalue : num [1:22883] 1.14e-08 1.70e-10 4.25e-07 3.60e-09 2.58e-09 ...
## ..- attr(*, ".internal.selfref")=<externalptr>
## ..- attr(*, "call")= chr "https://gemma.msl.ubc.ca/rest/v2/resultSets/535864"
## ..- attr(*, "env")=<environment: 0x10b2090f8>
## $ :Classes 'data.table' and 'data.frame': 30863 obs. of 11 variables:

```



```

## ..$ Probe : chr [1:30863] "1424921_at" "1451564_at" "1451777_at" "1435906_x_at"
## ..$ NCBIId : chr [1:30863] "69550" "547253" "234311" "14469" ...
## ..$ GeneSymbol : chr [1:30863] "Bst2" "Parp14" "Ddx60" "Gbp2" ...
## ..$ GeneName : chr [1:30863] "bone marrow stromal cell antigen 2" "poly (ADP-ribose)"
## ..$ pvalue : num [1:30863] 0 0 0 0 0 0 0 0 0 0 ...
## ..$ corrected_pvalue : num [1:30863] 0 0 0 0 0 0 0 0 0 0 ...
## ..$ rank : num [1:30863] 7e-04 7e-04 7e-04 7e-04 7e-04 7e-04 7e-04 7e-04 7e-04
## ..$ contrast_89756_coefficient: num [1:30863] 2.24 1.32 2.35 2.14 1.74 ...
## ..$ contrast_89756_log2fc : num [1:30863] 2.24 1.32 2.35 2.14 1.74 ...
## ..$ contrast_89756_tstat : num [1:30863] 12.01 8.9 12.97 9.51 9.34 ...
## ..$ contrast_89756_pvalue : num [1:30863] 3.67e-12 2.13e-09 6.51e-13 5.48e-10 8.08e-10 ...
## ..- attr(*, ".internal.selfref")=<externalptr>
## ..- attr(*, "call")= chr "https://gemma.msl.ubc.ca/rest/v2/resultSets/513873"
## ..- attr(*, "env")=<environment: 0x119a93ed0>
## $ :Classes 'data.table' and 'data.frame': 30868 obs. of 11 variables:
## ..$ Probe : chr [1:30868] "1418930_at" "1453196_a_at" "1450291_s_at" "1418191_s_at"
## ..$ NCBIId : chr [1:30868] "15945" "23962" "64380" "24110" ...
## ..$ GeneSymbol : chr [1:30868] "Cxc110" "Oas12" "Ms4a4c" "Usp18" ...
## ..$ GeneName : chr [1:30868] "C-X-C motif chemokine ligand 10" "2'-5' oligoadenylate synthetase"
## ..$ pvalue : num [1:30868] 8.55e-15 3.56e-14 1.44e-13 8.20e-13 6.62e-13 ...
## ..$ corrected_pvalue : num [1:30868] 3.85e-10 8.03e-10 2.17e-09 7.39e-09 7.39e-09 ...
## ..$ rank : num [1:30868] 2.22e-05 4.44e-05 6.66e-05 1.00e-04 8.88e-05 ...
## ..$ contrast_97695_coefficient: num [1:30868] 4.78 3.26 3.89 3.76 3.28 ...
## ..$ contrast_97695_log2fc : num [1:30868] 4.78 3.26 3.89 3.76 3.28 ...
## ..$ contrast_97695_tstat : num [1:30868] 34.7 30.1 32.6 24.3 26 ...
## ..$ contrast_97695_pvalue : num [1:30868] 6.64e-14 3.90e-13 1.43e-13 5.67e-12 2.37e-12 ...
## ..- attr(*, ".internal.selfref")=<externalptr>
## ..- attr(*, "call")= chr "https://gemma.msl.ubc.ca/rest/v2/resultSets/546199"
## ..- attr(*, "env")=<environment: 0x10e2897e8>
## $ :Classes 'data.table' and 'data.frame': 30890 obs. of 15 variables:
## ..$ Probe : chr [1:30890] "1434372_at" "1452428_a_at" "1418240_at" "1425156_at"
## ..$ NCBIId : chr [1:30890] "107350" "12010" "14469" "229900" ...
## ..$ GeneSymbol : chr [1:30890] "AW112010" "B2m" "Gbp2" "Gbp7" ...
## ..$ GeneName : chr [1:30890] "expressed sequence AW112010" "beta-2 microglobulin"
## ..$ pvalue : num [1:30890] 9.10e-07 3.19e-05 1.72e-05 7.98e-05 6.93e-05 ...
## ..$ corrected_pvalue : num [1:30890] 0.041 0.205 0.205 0.299 0.299 ...
## ..$ rank : num [1:30890] 2.22e-05 2.00e-04 6.65e-05 3.00e-04 2.00e-04 ...
## ..$ contrast_99431_coefficient: num [1:30890] 1.069 0.861 1.85 0.938 1.449 ...
## ..$ contrast_99431_log2fc : num [1:30890] 1.069 0.861 1.85 0.938 1.449 ...
## ..$ contrast_99431_tstat : num [1:30890] 19.7 12.9 13.4 10.2 10 ...
## ..$ contrast_99431_pvalue : num [1:30890] 1.11e-06 1.33e-05 1.07e-05 5.32e-05 5.67e-05 ...
## ..$ contrast_99432_coefficient: num [1:30890] -0.1711 0.1945 0.1047 0.0282 -0.0596 ...
## ..$ contrast_99432_log2fc : num [1:30890] -0.1711 0.1945 0.1047 0.0282 -0.0596 ...
## ..$ contrast_99432_tstat : num [1:30890] -3.152 2.915 0.758 0.305 -0.413 ...
## ..$ contrast_99432_pvalue : num [1:30890] 0.0198 0.0268 0.4774 0.7707 0.6941 ...
## ..- attr(*, ".internal.selfref")=<externalptr>
## ..- attr(*, "call")= chr "https://gemma.msl.ubc.ca/rest/v2/resultSets/466082"
## ..- attr(*, "env")=<environment: 0x10e1903c8>
## $ :Classes 'data.table' and 'data.frame': 21982 obs. of 15 variables:
## ..$ Probe : int [1:21982] 10462613 10420488 10389207 10496592 10399710 10444220
## ..$ NCBIId : chr [1:21982] "15958" "219132" "20304" "14469" ...
## ..$ GeneSymbol : chr [1:21982] "Ifit2" "Phf11d" "Cc15" "Gbp2" ...
## ..$ GeneName : chr [1:21982] "interferon-induced protein with tetratricopeptide repeat"
## ..$ pvalue : num [1:21982] 0 0 0 0 0 ...

```

```
## ..$ corrected_pvalue      : num [1:21982] 0 0 0 0 0 ...
## ..$ rank                  : num [1:21982] 1e-04 1e-04 1e-04 1e-04 1e-04 3e-04 3e-04 3e-04 3e-04
## ..$ contrast_140572_coefficient: num [1:21982] 5.01 4.71 5.57 5.32 4.91 ...
## ..$ contrast_140572_log2fc   : num [1:21982] 5.01 4.71 5.57 5.32 4.91 ...
## ..$ contrast_140572_tstat    : num [1:21982] 87 68.3 67.3 71.6 66.8 ...
## ..$ contrast_140572_pvalue   : num [1:21982] 0 0 0 0 0 ...
## ..$ contrast_140573_coefficient: num [1:21982] 4.85 4.02 5.53 5.3 5.03 ...
## ..$ contrast_140573_log2fc   : num [1:21982] 4.85 4.02 5.53 5.3 5.03 ...
## ..$ contrast_140573_tstat    : num [1:21982] 84.2 58.3 66.8 71.3 68.4 ...
## ..$ contrast_140573_pvalue   : num [1:21982] 0.00 4.44e-16 0.00 0.00 0.00 ...
## ..- attr(*, ".internal.selfref")=<externalptr>
## ..- attr(*, "call")= chr "https://gemma.msl.ubc.ca/rest/v2/resultSets/522293"
## ..- attr(*, "env")=<environment: 0x10e190eb8>
## $ :Classes 'data.table' and 'data.frame': 19466 obs. of 11 variables:
## ..$ Probe                  : chr [1:19466] "1451777_at" "1418293_at" "1419282_at" "1418825_at"
## ..$ NCBIId                 : chr [1:19466] "234311" "15958" "20293" "15944" ...
## ..$ GeneSymbol             : chr [1:19466] "Ddx60" "Ifit2" "Ccl12" "Irgm1" ...
## ..$ GeneName               : chr [1:19466] "DEXD/H box helicase 60" "interferon-induced protein
## ..$ pvalue                 : num [1:19466] 0 0 0 0 0 0 0 0 0 ...
## ..$ corrected_pvalue      : num [1:19466] 0 0 0 0 0 0 0 0 0 ...
## ..$ rank                   : num [1:19466] 0.0015 0.0015 0.0015 0.0015 0.0015 0.0015 0.0015 0.0015 0.
## ..$ contrast_142941_coefficient: num [1:19466] 4.85 5.86 5.61 4.39 5.08 ...
## ..$ contrast_142941_log2fc   : num [1:19466] 4.85 5.86 5.61 4.39 5.08 ...
## ..$ contrast_142941_tstat    : num [1:19466] 51.5 86.3 56.7 62.5 55.7 ...
## ..$ contrast_142941_pvalue   : num [1:19466] 0 0 0 0 0 0 0 0 0 ...
## ..- attr(*, ".internal.selfref")=<externalptr>
## ..- attr(*, "call")= chr "https://gemma.msl.ubc.ca/rest/v2/resultSets/523663"
## ..- attr(*, "env")=<environment: 0x10e170550>
```

```
# Create a function to save DE results for each ResultSet
```

```
SavingGemmaDEResults_forEachResultSet <- function(differentials,
                                                    UniqueResultSetIDs,
                                                    ResultSets_contrasts){

  # Loop through each dataset (list element) in "differentials" object
  for (i in c(1:length(differentials))){

    # Get the current ResultSet ID
    ThisResultSet <- UniqueResultSetIDs[i]

    # Get the dataset ID corresponding to the current ResultSet ID
    # Some datasets have multiple ResultSets, so take the dataset ID from the
    # first matching entry
    ThisDataSet <- ResultSets_contrasts$ExperimentID[ResultSets_contrasts$ResultSetIDs == ThisResultSet]

    # Export the DE results for the current ResultSet ID into the working directory
    write.csv(differentials[[i]], paste("DEResults_", ThisDataSet, "_", ThisResultSet, ".csv", sep=""))

    # Remove the temporary IDs
    rm(ThisDataSet, ThisResultSet)
  }

  # Print a message to indicate the DE results have been exported into working directory
  print("Output object: DEResults_Dataset ID_ResultSet ID")
}
```

```
}
```

```
# Function use  
SavingGemmaDEResults_forEachResultSet(differentials,  
                                       UniqueResultSetIDs,  
                                       ResultSets_contrasts)
```

```
## [1] "Output object: DEResults_Dataset ID_ResultSet ID"
```

6) Filter of DE results for rows with good gene annotation

```
# Create a function to filter for rows with good gene annotation  
FilteringDEResults_GoodAnnotation <- function(DE_Results){  
  
  # Print the total number of rows in the DE results  
  print("# of rows in results")  
  print(nrow(DE_Results))  
  
  # Print the number of rows with missing NCBI annotation  
  print("# of rows with missing NCBI annotation:")  
  print(sum(DE_Results$NCBIid == "" | DE_Results$NCBIid == "null", na.rm = TRUE))  
  
  # Print the number of rows with NA NCBI annotation  
  print("# of rows with NA NCBI annotation:")  
  print(sum(is.na(DE_Results$NCBIid)))  
  
  # Print the number of rows with missing Gene Symbol annotation  
  print("# of rows with missing Gene Symbol annotation:")  
  print(sum(DE_Results$GeneSymbol == "" | DE_Results$GeneSymbol == "null", na.rm = TRUE))  
  
  # Print the number of rows mapped to multiple NCBI IDs  
  print("# of rows mapped to multiple NCBI IDs:")  
  print(length(grep('\\|', DE_Results$NCBIid)))  
  
  # Print the number of rows mapped to multiple Gene Symbols  
  print("# of rows mapped to multiple Gene Symbols:")  
  print(length(grep('\\|', DE_Results$GeneSymbol)))  
  
  # Subset data containing rows with valid NCBI EntrezID (non-empty and non-null)  
  DE_Results_NoNA <- DE_Results[(DE_Results$NCBIid == "" |  
                                DE_Results$NCBIid == "null") == FALSE &  
                                is.na(DE_Results$NCBIid) == FALSE,]  
  
  # Subset data annotated with a single gene (not ambiguously mapped to more  
  # than one gene)  
  if(length(grep('\\|', DE_Results_NoNA$NCBIid)) == 0){  
    # If there are no rows with multiple NCBI IDs, use the current subset  
    DE_Results_GoodAnnotation <- DE_Results_NoNA  
  } else {  
    # Extract only rows annotated with a single Gene Symbol (no pipe character '|')  
    DE_Results_GoodAnnotation <- DE_Results_NoNA[~(grep('\\|', DE_Results_NoNA$NCBIid)),]
```

```

}

# Print the number of rows with good annotation
print("# of rows with good annotation")
print(nrow(DE_Results_GoodAnnotation))

# Get the name of the input object as a string for file naming
ID <- deparse(substitute(DE_Results))

# Export the DE results with good annotations into the working directory
write.csv(DE_Results_GoodAnnotation, paste(ID, "_GoodAnnotation.csv", sep = ""))

# Remove the temporary DE result objects
rm(DE_Results_NoNA, DE_Results)

# Print a message to the DE results with good annotations have been exported
# into working directory
print(paste("Output object:", ID, "_GoodAnnotation.csv", sep = ""))

# Return the DE results with good annotation into the environment
return(DE_Results_GoodAnnotation)
}

# Separate the DE results from object "differentials"
DEResults_GSE30577 <- differentials[[1]]
DEResults_GSE42264 <- differentials[[2]]
DEResults_GSE44331 <- differentials[[3]]
DEResults_GSE51365 <- differentials[[4]]
DEResults_GSE53784 <- differentials[[5]]
DEResults_GSE91074 <- differentials[[6]]

# Function use
DE_Results_GSE30577_GoodAnnotation <- FilteringDEResults_GoodAnnotation(DEResults_GSE30577)

## [1] "# of rows in results"
## [1] 22883
## [1] "# of rows with missing NCBI annotation:"
## [1] 0
## [1] "# of rows with NA NCBI annotation:"
## [1] 0
## [1] "# of rows with missing Gene Symbol annotation:"
## [1] 0
## [1] "# of rows mapped to multiple NCBI IDs:"
## [1] 0
## [1] "# of rows mapped to multiple Gene Symbols:"
## [1] 0
## [1] "# of rows with good annotation"
## [1] 22883
## [1] "Output object:DEResults_GSE30577_GoodAnnotation.csv"

DE_Results_GSE42264_GoodAnnotation <- FilteringDEResults_GoodAnnotation(DEResults_GSE42264)

## [1] "# of rows in results"

```

```
## [1] 30863
## [1] "# of rows with missing NCBI annotation:"
## [1] 0
## [1] "# of rows with NA NCBI annotation:"
## [1] 0
## [1] "# of rows with missing Gene Symbol annotation:"
## [1] 0
## [1] "# of rows mapped to multiple NCBI IDs:"
## [1] 0
## [1] "# of rows mapped to multiple Gene Symbols:"
## [1] 0
## [1] "# of rows with good annotation"
## [1] 30863
## [1] "Output object:DEResults_GSE42264_GoodAnnotation.csv"
```

```
DE_Results_GSE44331_GoodAnnotation <- FilteringDEResults_GoodAnnotation(DEResults_GSE44331)
```

```
## [1] "# of rows in results"
## [1] 30868
## [1] "# of rows with missing NCBI annotation:"
## [1] 0
## [1] "# of rows with NA NCBI annotation:"
## [1] 0
## [1] "# of rows with missing Gene Symbol annotation:"
## [1] 0
## [1] "# of rows mapped to multiple NCBI IDs:"
## [1] 0
## [1] "# of rows mapped to multiple Gene Symbols:"
## [1] 0
## [1] "# of rows with good annotation"
## [1] 30868
## [1] "Output object:DEResults_GSE44331_GoodAnnotation.csv"
```

```
DE_Results_GSE51365_GoodAnnotation <- FilteringDEResults_GoodAnnotation(DEResults_GSE51365)
```

```
## [1] "# of rows in results"
## [1] 30890
## [1] "# of rows with missing NCBI annotation:"
## [1] 0
## [1] "# of rows with NA NCBI annotation:"
## [1] 0
## [1] "# of rows with missing Gene Symbol annotation:"
## [1] 0
## [1] "# of rows mapped to multiple NCBI IDs:"
## [1] 0
## [1] "# of rows mapped to multiple Gene Symbols:"
## [1] 0
## [1] "# of rows with good annotation"
## [1] 30890
## [1] "Output object:DEResults_GSE51365_GoodAnnotation.csv"
```

```
DE_Results_GSE53784_GoodAnnotation <- FilteringDEResults_GoodAnnotation(DEResults_GSE53784)
```

```
## [1] "# of rows in results"
## [1] 21982
## [1] "# of rows with missing NCBI annotation:"
## [1] 0
## [1] "# of rows with NA NCBI annotation:"
## [1] 0
## [1] "# of rows with missing Gene Symbol annotation:"
## [1] 0
## [1] "# of rows mapped to multiple NCBI IDs:"
## [1] 0
## [1] "# of rows mapped to multiple Gene Symbols:"
## [1] 0
## [1] "# of rows with good annotation"
## [1] 21982
## [1] "Output object:DEResults_GSE53784_GoodAnnotation.csv"
```

```
DE_Results_GSE91074_GoodAnnotation <- FilteringDEResults_GoodAnnotation(DEResults_GSE91074)
```

```
## [1] "# of rows in results"
## [1] 19466
## [1] "# of rows with missing NCBI annotation:"
## [1] 0
## [1] "# of rows with NA NCBI annotation:"
## [1] 0
## [1] "# of rows with missing Gene Symbol annotation:"
## [1] 0
## [1] "# of rows mapped to multiple NCBI IDs:"
## [1] 0
## [1] "# of rows mapped to multiple Gene Symbols:"
## [1] 0
## [1] "# of rows with good annotation"
## [1] 19466
## [1] "Output object:DEResults_GSE91074_GoodAnnotation.csv"
```

```
# Check the structure of the DE results with good annotations
# str(DE_Results_GSE30577_GoodAnnotation)
# str(DE_Results_GSE42264_GoodAnnotation)
# str(DE_Results_GSE44331_GoodAnnotation)
# str(DE_Results_GSE51365_GoodAnnotation)
# str(DE_Results_GSE53784_GoodAnnotation)
# str(DE_Results_GSE91074_GoodAnnotation)
```

7) Extraction of DE results for the contrasts of interest

```
# Create a function to extract the contrast ID from FC column names
GetContrastIDsforResultSet <- function(NamesOfFoldChangeColumns){

  # Split the column names using the underscore as a delimiter
```

```

# The result is a list where each element is a vector of the split parts of
# each column name
ColumnNames_BrokenUp <- strsplit(NamesOfFoldChangeColumns, "_")

# Convert the list of split names to a matrix
MatrixOfColumnNames_BrokenUp <- do.call(rbind, ColumnNames_BrokenUp)

# Extract the contrast IDs in the second column
ContrastIDs_inCurrentDF <- MatrixOfColumnNames_BrokenUp[, 2]

# Return the extracted contrast ID
return(ContrastIDs_inCurrentDF)
}

# Create a function to extract DE results for contrasts of interest
ExtractingDEResultsForContrasts <- function(DE_Results_GoodAnnotation,
                                             Contrasts_Log2FC,
                                             Contrasts_Tstat,
                                             ResultSet_contrasts){

  # Print the column names in the DE results with good annotations for the
  # current ResultSet
  print("Columns in the DE results for the current ResultSet:")
  print(colnames(DE_Results_GoodAnnotation))

  # Print the column names that correspond to Log2FC values for contrasts of interest
  print("Columns of Log2FC for contrasts of interest within the DE results for the current ResultSet:")
  NamesOfFoldChangeColumns <- colnames(DE_Results_GoodAnnotation)[colnames(DE_Results_GoodAnnotation) %in%
  print(NamesOfFoldChangeColumns)

  # Print the column names that correspond to T-statistics values for contrasts of interest
  print("Columns of T-statistics for contrasts of interest within the DE results for the current ResultSet:")
  NamesOfTstatColumns <- colnames(DE_Results_GoodAnnotation)[colnames(DE_Results_GoodAnnotation) %in%
  print(NamesOfTstatColumns)

  # Extract contrast IDs from FC column names using the function "GetContrastIDsforResultSet"
  ContrastIDs_inCurrentDF <- GetContrastIDsforResultSet(NamesOfFoldChangeColumns)
  print("Contrast IDs for contrasts of interest within the current ResultSet:")
  print(ContrastIDs_inCurrentDF)

  # Extract dataset IDs for contrasts of interest
  DatasetIDs <- ResultSet_contrasts$ExperimentID[ResultSet_contrasts$ContrastIDs %in% ContrastIDs_inCurrentDF]
  print("Dataset ID for the ResultSet and contrasts:")
  print(DatasetIDs)

  # Extract experimental factors for contrasts of interest
  Factors_inCurrentDF <- ResultSet_contrasts$ExperimentalFactors[ResultSet_contrasts$ContrastIDs %in% ContrastIDs_inCurrentDF]
  print("Experimental factors for ResultSet and contrasts:")
  print(Factors_inCurrentDF)

  # Combine dataset IDs and experimental factors to create unique identifiers
  # for each statistical comparison
  ComparisonsOfInterest <- paste(DatasetIDs, Factors_inCurrentDF, sep = "_")

```

```

print("Current names of contrasts of interest")
print(ComparisonsOfInterest)

# Create a list to store extracted DE results and relevant metadata
DE_result_contrast <- list(
  All_Columns = colnames(DE_Results_GoodAnnotation),
  NamesOfFoldChangeColumns = NamesOfFoldChangeColumns,
  NamesOfTstatColumns = NamesOfTstatColumns,
  Contrast_ID = ContrastIDs_inCurrentDF,
  Dataset_ID = DatasetIDs,
  Experimental_Factor = Factors_inCurrentDF,
  ComparisonsOfInterest = ComparisonsOfInterest
)

# Return the list of all extracted info
return(DE_result_contrast)
}

# Create a function to extract the Log2FC and T-statistics of contrast of interest
GetContrastStatColumns <- function(ResultSets_contrasts){

  # Extract the effect sizes (Log2FC) of contrasts of interest
  Contrasts_Log2FC <- paste("contrast_", ResultSets_contrasts$ContrastIDs, "_log2fc", sep = "")

  # Extract the T-statistics of contrasts of interest for calculating sampling variances
  Contrasts_Tstat <- paste("contrast_", ResultSets_contrasts$ContrastIDs, "_tstat", sep = "")

  # Create a dataframe to store the extracted information
  Contrasts_Stat <- data.frame(
    ExperimentID = ResultSets_contrasts$ExperimentIDs,
    ContrastID = ResultSets_contrasts$ContrastIDs,
    Log2FC_Column = Contrasts_Log2FC,
    Tstat_Column = Contrasts_Tstat,
    stringsAsFactors = FALSE
  )

  # Return the resulting dataframe
  return(Contrasts_Stat)
}

# Function use
Contrasts_Stat_Columns <- GetContrastStatColumns(ResultSets_contrasts)

# Prepare "Contrast_Log2FC" and "Contrast_Tstat" arguments for extracting DE results
# of contrasts of interest

# GSE30577
Contrasts_Log2FC_GSE30577 <- Contrasts_Stat_Columns %>%
  filter(ExperimentID == "GSE30577") %>%
  pull(Log2FC_Column)

Contrasts_Tstat_GSE30577 <- Contrasts_Stat_Columns %>%
  filter(ExperimentID == "GSE30577") %>%
  pull(Tstat_Column)

```



```

# GSE42264
Contrasts_Log2FC_GSE42264 <- Contrasts_Stat_Columns %>%
  filter(ExperimentID == "GSE42264") %>%
  pull(Log2FC_Column)

Contrasts_Tstat_GSE42264 <- Contrasts_Stat_Columns %>%
  filter(ExperimentID == "GSE42264") %>%
  pull(Tstat_Column)

# GSE44331
Contrasts_Log2FC_GSE44331 <- Contrasts_Stat_Columns %>%
  filter(ExperimentID == "GSE44331") %>%
  pull(Log2FC_Column)

Contrasts_Tstat_GSE44331 <- Contrasts_Stat_Columns %>%
  filter(ExperimentID == "GSE44331") %>%
  pull(Tstat_Column)

# GSE51365
Contrasts_Log2FC_GSE51365 <- Contrasts_Stat_Columns %>%
  filter(ExperimentID == "GSE51365") %>%
  pull(Log2FC_Column)

Contrasts_Tstat_GSE51365 <- Contrasts_Stat_Columns %>%
  filter(ExperimentID == "GSE51365") %>%
  pull(Tstat_Column)

# GSE53784
Contrasts_Log2FC_GSE53784 <- Contrasts_Stat_Columns %>%
  filter(ExperimentID == "GSE53784") %>%
  pull(Log2FC_Column)

Contrasts_Tstat_GSE53784 <- Contrasts_Stat_Columns %>%
  filter(ExperimentID == "GSE53784") %>%
  pull(Tstat_Column)

# GSE91074
Contrasts_Log2FC_GSE91074 <- Contrasts_Stat_Columns %>%
  filter(ExperimentID == "GSE91074") %>%
  pull(Log2FC_Column)

Contrasts_Tstat_GSE91074 <- Contrasts_Stat_Columns %>%
  filter(ExperimentID == "GSE91074") %>%
  pull(Tstat_Column)

# Function use

# GSE30577
DE_Results_Contrasts_GSE30577 <- ExtractingDEResultsForContrasts(DE_Results_GSE30577_GoodAnnotation,
  Contrasts_Log2FC_GSE30577,
  Contrasts_Tstat_GSE30577,
  ResultSets_contrasts)

```

```
## [1] "Columns in the DE results for the current ResultSet:"
## [1] "Probe" "NCBIid"
## [3] "GeneSymbol" "GeneName"
## [5] "pvalue" "corrected_pvalue"
## [7] "rank" "contrast_84556_coefficient"
## [9] "contrast_84556_log2fc" "contrast_84556_tstat"
## [11] "contrast_84556_pvalue"
## [1] "Columns of Log2FC for contrasts of interest within the DE results for the current ResultSet:"
## [1] "contrast_84556_log2fc"
## [1] "Columns of T-statistics for contrasts of interest within the DE results for the current ResultSet:"
## [1] "contrast_84556_tstat"
## [1] "Contrast IDs for contrasts of interest within the current ResultSet:"
## [1] "84556"
## [1] "Dataset ID for the ResultSet and contrasts:"
## [1] "GSE30577"
## [1] "Experimental factors for ResultSet and contrasts:"
## [1] "Rabies virus has_genotype strain CVS-11"
## [1] "Current names of contrasts of interest"
## [1] "GSE30577_Rabies virus has_genotype strain CVS-11"
```

GSE42264

```
DE_Results_Contrasts_GSE42264 <- ExtractingDEResultsForContrasts(DE_Results_GSE42264_GoodAnnotation,
                                                                    Contrasts_Log2FC_GSE42264,
                                                                    Contrasts_Tstat_GSE42264,
                                                                    ResultSets_contrasts)
```

```
## [1] "Columns in the DE results for the current ResultSet:"
## [1] "Probe" "NCBIid"
## [3] "GeneSymbol" "GeneName"
## [5] "pvalue" "corrected_pvalue"
## [7] "rank" "contrast_89756_coefficient"
## [9] "contrast_89756_log2fc" "contrast_89756_tstat"
## [11] "contrast_89756_pvalue"
## [1] "Columns of Log2FC for contrasts of interest within the DE results for the current ResultSet:"
## [1] "contrast_89756_log2fc"
## [1] "Columns of T-statistics for contrasts of interest within the DE results for the current ResultSet:"
## [1] "contrast_89756_tstat"
## [1] "Contrast IDs for contrasts of interest within the current ResultSet:"
## [1] "89756"
## [1] "Dataset ID for the ResultSet and contrasts:"
## [1] "GSE42264"
## [1] "Experimental factors for ResultSet and contrasts:"
## [1] "measles"
## [1] "Current names of contrasts of interest"
## [1] "GSE42264_measles"
```

GSE44331

```
DE_Results_Contrasts_GSE44331 <- ExtractingDEResultsForContrasts(DE_Results_GSE44331_GoodAnnotation,
                                                                    Contrasts_Log2FC_GSE44331,
                                                                    Contrasts_Tstat_GSE44331,
                                                                    ResultSets_contrasts)
```

```
## [1] "Columns in the DE results for the current ResultSet:"
```

```
## [1] "Probe" "NCBIid"
## [3] "GeneSymbol" "GeneName"
## [5] "pvalue" "corrected_pvalue"
## [7] "rank" "contrast_97695_coefficient"
## [9] "contrast_97695_log2fc" "contrast_97695_tstat"
## [11] "contrast_97695_pvalue"
## [1] "Columns of Log2FC for contrasts of interest within the DE results for the current ResultSet:"
## [1] "contrast_97695_log2fc"
## [1] "Columns of T-statistics for contrasts of interest within the DE results for the current ResultSet:"
## [1] "contrast_97695_tstat"
## [1] "Contrast IDs for contrasts of interest within the current ResultSet:"
## [1] "97695"
## [1] "Dataset ID for the ResultSet and contrasts:"
## [1] "GSE44331"
## [1] "Experimental factors for ResultSet and contrasts:"
## [1] "Vesicular stomatitis Indiana virus"
## [1] "Current names of contrasts of interest"
## [1] "GSE44331_Vesicular stomatitis Indiana virus"
```

```
# GSE51365
```

```
DE_Results_Contrasts_GSE51365 <- ExtractingDEResultsForContrasts(DE_Results_GSE51365_GoodAnnotation,
                                                                    Contrasts_Log2FC_GSE51365,
                                                                    Contrasts_Tstat_GSE51365,
                                                                    ResultSets_contrasts)
```

```
## [1] "Columns in the DE results for the current ResultSet:"
## [1] "Probe" "NCBIid"
## [3] "GeneSymbol" "GeneName"
## [5] "pvalue" "corrected_pvalue"
## [7] "rank" "contrast_99431_coefficient"
## [9] "contrast_99431_log2fc" "contrast_99431_tstat"
## [11] "contrast_99431_pvalue" "contrast_99432_coefficient"
## [13] "contrast_99432_log2fc" "contrast_99432_tstat"
## [15] "contrast_99432_pvalue"
## [1] "Columns of Log2FC for contrasts of interest within the DE results for the current ResultSet:"
## [1] "contrast_99431_log2fc" "contrast_99432_log2fc"
## [1] "Columns of T-statistics for contrasts of interest within the DE results for the current ResultSet:"
## [1] "contrast_99431_tstat" "contrast_99432_tstat"
## [1] "Contrast IDs for contrasts of interest within the current ResultSet:"
## [1] "99431" "99432"
## [1] "Dataset ID for the ResultSet and contrasts:"
## [1] "GSE51365" "GSE51365"
## [1] "Experimental factors for ResultSet and contrasts:"
## [1] "murine gammaherpesvirus 68"
## [2] "murine gammaherpesvirus 68 has phenotype latency-defective"
## [1] "Current names of contrasts of interest"
## [1] "GSE51365_murine gammaherpesvirus 68"
## [2] "GSE51365_murine gammaherpesvirus 68 has phenotype latency-defective"
```

```
# GSE53784
```

```
DE_Results_Contrasts_GSE53784 <- ExtractingDEResultsForContrasts(DE_Results_GSE53784_GoodAnnotation,
                                                                    Contrasts_Log2FC_GSE53784,
                                                                    Contrasts_Tstat_GSE53784,
                                                                    ResultSets_contrasts)
```

```
## [1] "Columns in the DE results for the current ResultSet:"
## [1] "Probe" "NCBIId"
## [3] "GeneSymbol" "GeneName"
## [5] "pvalue" "corrected_pvalue"
## [7] "rank" "contrast_140572_coefficient"
## [9] "contrast_140572_log2fc" "contrast_140572_tstat"
## [11] "contrast_140572_pvalue" "contrast_140573_coefficient"
## [13] "contrast_140573_log2fc" "contrast_140573_tstat"
## [15] "contrast_140573_pvalue"
## [1] "Columns of Log2FC for contrasts of interest within the DE results for the current ResultSet:"
## [1] "contrast_140572_log2fc" "contrast_140573_log2fc"
## [1] "Columns of T-statistics for contrasts of interest within the DE results for the current ResultSet:"
## [1] "contrast_140572_tstat" "contrast_140573_tstat"
## [1] "Contrast IDs for contrasts of interest within the current ResultSet:"
## [1] "140572" "140573"
## [1] "Dataset ID for the ResultSet and contrasts:"
## [1] "GSE53784" "GSE53784"
## [1] "Experimental factors for ResultSet and contrasts:"
## [1] "West Nile virus" "Japanese encephalitis virus"
## [1] "Current names of contrasts of interest"
## [1] "GSE53784_West Nile virus"
## [2] "GSE53784_Japanese encephalitis virus"
```

```
# GSE91074
```

```
DE_Results_Contrasts_GSE91074 <- ExtractingDEResultsForContrasts(DE_Results_GSE91074_GoodAnnotation,
                                                                    Contrasts_Log2FC_GSE91074,
                                                                    Contrasts_Tstat_GSE91074,
                                                                    ResultSets_contrasts)
```

```
## [1] "Columns in the DE results for the current ResultSet:"
## [1] "Probe" "NCBIId"
## [3] "GeneSymbol" "GeneName"
## [5] "pvalue" "corrected_pvalue"
## [7] "rank" "contrast_142941_coefficient"
## [9] "contrast_142941_log2fc" "contrast_142941_tstat"
## [11] "contrast_142941_pvalue"
## [1] "Columns of Log2FC for contrasts of interest within the DE results for the current ResultSet:"
## [1] "contrast_142941_log2fc"
## [1] "Columns of T-statistics for contrasts of interest within the DE results for the current ResultSet:"
## [1] "contrast_142941_tstat"
## [1] "Contrast IDs for contrasts of interest within the current ResultSet:"
## [1] "142941"
## [1] "Dataset ID for the ResultSet and contrasts:"
## [1] "GSE91074"
## [1] "Experimental factors for ResultSet and contrasts:"
## [1] "Venezuelan equine encephalitis virus strain TC83"
## [1] "Current names of contrasts of interest"
## [1] "GSE91074_Venezuelan equine encephalitis virus strain TC83"
```

8) Collapse of DE results to one result per gene & Calculation of standard error

and sampling variance

Gene expression can be measured using multiple probes (microarray). Therefore, DE results need to be collapsed to one result per gene.

Standard error of $\text{Log2FC} = \text{Log2FC}/T\text{-statistics}$

Sampling variance = Average of standard error of each gene²

```
CollapsingDEResults_OneResultPerGene<-function(GSE_ID,
                                              DE_Results_GoodAnnotation,
                                              ComparisonsOfInterest,
                                              NamesOfFoldChangeColumns,
                                              NamesOfTstatColumns){

  # Print a message to check if the vectors containing FC and T-stat column names
  # are in the same order as the comparisons of interest
  print("Check if the vectors containing FC and Tstat column names contain the same order as the compar")

  # Print the number of unique NCBI IDs in the DE results
  print("# of rows with unique NCBI IDs:")
  print(length(unique(DE_Results_GoodAnnotation$NCBIId)))

  # Print the number of unique Gene Symbols in the DE results
  print("# of rows with unique Gene Symbols:")
  print(length(unique(DE_Results_GoodAnnotation$GeneSymbol)))

  # Create a folder named after the dataset ID to store results
  dir.create(paste("./", "Collapsing_DEResults_", GSE_ID, sep=""))

  # Set the working directory to the newly created folder
  setwd(paste("./", "Collapsing_DEResults_", GSE_ID, sep=""))

  # Create lists to store results
  DE_Results_GoodAnnotation_FoldChange_Average <- list()
  DE_Results_GoodAnnotation_Tstat_Average <- list()
  DE_Results_GoodAnnotation_SE_Average <- list()

  # Loop through each column containing FC and T-statistic info for the contrasts of interest
  for(i in c(1:length(NamesOfFoldChangeColumns))){

    # Select the Log2FC column of interest
    FoldChangeColumn <- select(DE_Results_GoodAnnotation, NamesOfFoldChangeColumns[i])

    # Select the T-stat column of interest
    TstatColumn <- select(DE_Results_GoodAnnotation, NamesOfTstatColumns[i])

    # Calculate the standard error (SE)
    DE_Results_GoodAnnotation_SE <- FoldChangeColumn[[1]]/TstatColumn[[1]]

    # Calculate the average Log2FC per gene
    DE_Results_GoodAnnotation_FoldChange_Average[[i]] <- tapply(FoldChangeColumn[[1]],
                                                                DE_Results_GoodAnnotation$NCBIId,
```

```

mean)

# Calculate the average T-statistics per gene
DE_Results_GoodAnnotation_Tstat_Average[[i]] <- tapply(TstatColumn[[1]],
DE_Results_GoodAnnotation$NCBIId,
mean)

# Calculate the average SE per gene
DE_Results_GoodAnnotation_SE_Average[[i]] <- tapply(DE_Results_GoodAnnotation_SE,
DE_Results_GoodAnnotation$NCBIId,
mean)

}

# Combine averaged Log2FC values into a single dataframe
DE_Results_GoodAnnotation_FoldChange_AveragedByGene <- do.call(cbind, DE_Results_GoodAnnotation_FoldC

# Print the dimensions of the averaged Fold Change matrix
print("Dimensions of Fold Change matrix, averaged by gene symbol:")
print(dim(DE_Results_GoodAnnotation_FoldChange_AveragedByGene))

# Name the columns in the dataframe to describe the contrast of interest
colnames(DE_Results_GoodAnnotation_FoldChange_AveragedByGene) <- ComparisonsOfInterest

# Export the averaged Log2FC results into the working directory
write.csv(DE_Results_GoodAnnotation_FoldChange_AveragedByGene,
paste("DEResults_", GSE_ID, "_GoodAnnotation_FoldChange_AveragedByGene.csv", sep = ""))

# Combine T-statistics values into a single dataframe
DE_Results_GoodAnnotation_Tstat_AveragedByGene<-do.call(cbind, DE_Results_GoodAnnotation_Tstat_Averag

# Name the columns in the dataframe to describe the contrast of interest
colnames(DE_Results_GoodAnnotation_Tstat_AveragedByGene) <- ComparisonsOfInterest

# Export the averaged T-statistics results into the working directory
write.csv(DE_Results_GoodAnnotation_Tstat_AveragedByGene,
paste("DEResults_", GSE_ID, "_GoodAnnotation_Tstat_AveragedByGene.csv", sep = ""))

# Combine SE values into a single dataframe
DE_Results_GoodAnnotation_SE_AveragedByGene<-do.call(cbind, DE_Results_GoodAnnotation_SE_Average)

# Name the columns in the dataframe to describe the contrast of interest
colnames(DE_Results_GoodAnnotation_SE_AveragedByGene) <- ComparisonsOfInterest

# Export the averaged SE results into the working directory
write.csv(DE_Results_GoodAnnotation_SE_AveragedByGene,
paste("DEResults_", GSE_ID, "_GoodAnnotation_SE_AveragedByGene.csv", sep = ""))

# Calculate the sampling variance (SV) by squaring the SE
DE_Results_GoodAnnotation_SV <- (DE_Results_GoodAnnotation_SE_AveragedByGene)^2

# Export the SV results into the working directory
write.csv(DE_Results_GoodAnnotation_SV,

```

```

    paste("DEResults_", GSE_ID, "_GoodAnnotation_SV.csv", sep = "")

# Compile all averaged results into a single list
TempMasterResults<-list(Log2FC = DE_Results_GoodAnnotation_FoldChange_AveragedByGene,
                        Tstat = DE_Results_GoodAnnotation_Tstat_AveragedByGene,
                        SE = DE_Results_GoodAnnotation_SE_AveragedByGene,
                        SV = DE_Results_GoodAnnotation_SV)

# Print the name of the output
print(paste("Output: Collapsing_DEResults", GSE_ID, sep="_"))

# Clean up the environment by removing temporary results
rm(DE_Results_GoodAnnotation, DE_Results_GoodAnnotation_SV,
   DE_Results_GoodAnnotation_SE, DE_Results_GoodAnnotation_FoldChange_AveragedByGene,
   DE_Results_GoodAnnotation_FoldChange_Average, DE_Results_GoodAnnotation_Tstat_AveragedByGene,
   DE_Results_GoodAnnotation_Tstat_Average, DE_Results_GoodAnnotation_SE_Average,
   FoldChangeColumn, TstatColumn, GSE_ID, ComparisonsOfInterest, NamesOfFoldChangeColumns,
   NamesOfTstatColumns)

# Set the working directory back to the parent directory
setwd("../")

# Return the compiled results
return(TempMasterResults)
}

```

```

# Prepare "NamesOfFoldChangeColumns" and "NamesOfTstatColumns" arguments for
# extracting DE results of contrasts of interest

# GSE30577
NamesOfFoldChangeColumns_GSE30577 <- DE_Results_Contrasts_GSE30577[[2]]
NamesOfTstatColumns_GSE30577 <- DE_Results_Contrasts_GSE30577[[3]]
ComparisonsOfInterest_GSE30577 <- c("GSE30577_Rabies_vs_Control")

# GSE42264
NamesOfFoldChangeColumns_GSE42264 <- DE_Results_Contrasts_GSE42264[[2]]
NamesOfTstatColumns_GSE42264 <- DE_Results_Contrasts_GSE42264[[3]]
ComparisonsOfInterest_GSE42264 <- c("GSE42264_Measles_vs_Control")

# GSE44331
NamesOfFoldChangeColumns_GSE44331 <- DE_Results_Contrasts_GSE44331[[2]]
NamesOfTstatColumns_GSE44331 <- DE_Results_Contrasts_GSE44331[[3]]
ComparisonsOfInterest_GSE44331 <- c("GSE44331_VSV_vs_Control")

# GSE51365
NamesOfFoldChangeColumns_GSE51365 <- DE_Results_Contrasts_GSE51365[[2]]
NamesOfTstatColumns_GSE51365 <- DE_Results_Contrasts_GSE51365[[3]]
ComparisonsOfInterest_GSE51365 <- c("GSE51365_MVH68_WT_vs_Control", "GSE51365_MVH68_M_vs_Control")

# GSE53784
NamesOfFoldChangeColumns_GSE53784 <- DE_Results_Contrasts_GSE53784[[2]]
NamesOfTstatColumns_GSE53784 <- DE_Results_Contrasts_GSE53784[[3]]
ComparisonsOfInterest_GSE53784 <- c("GSE53784_WNV_vs_Control", "GSE53784_JEV_vs_Control")

```



```

# GSE91074
NamesOfFoldChangeColumns_GSE91074 <- DE_Results_Contrasts_GSE91074[[2]]
NamesOfTstatColumns_GSE91074 <- DE_Results_Contrasts_GSE91074[[3]]
ComparisonsOfInterest_GSE91074 <- c("GSE91074_VEEV_vs_Control")

# Function use

# GSE30577
Collapsing_DEResults_GSE30577 <- CollapsingDEResults_OneResultPerGene("GSE30577",
                                                                    DE_Results_GSE30577_GoodAnnotation,
                                                                    ComparisonsOfInterest_GSE30577,
                                                                    NamesOfFoldChangeColumns_GSE30577,
                                                                    NamesOfTstatColumns_GSE30577)

## [1] "Check if the vectors containing FC and Tstat column names contain the same order as the comparison"
## [1] "# of rows with unique NCBI IDs:"
## [1] 16125
## [1] "# of rows with unique Gene Symbols:"
## [1] 16125

## Warning in dir.create(paste("./", "Collapsing_DEResults_", GSE_ID, sep = "")):
## './Collapsing_DEResults_GSE30577' already exists

## [1] "Dimensions of Fold Change matrix, averaged by gene symbol:"
## [1] 16125      1
## [1] "Output: Collapsing_DEResults_GSE30577"

# GSE42264
Collapsing_DEResults_GSE42264 <- CollapsingDEResults_OneResultPerGene("GSE42264",
                                                                    DE_Results_GSE42264_GoodAnnotation,
                                                                    ComparisonsOfInterest_GSE42264,
                                                                    NamesOfFoldChangeColumns_GSE42264,
                                                                    NamesOfTstatColumns_GSE42264)

## [1] "Check if the vectors containing FC and Tstat column names contain the same order as the comparison"
## [1] "# of rows with unique NCBI IDs:"
## [1] 18555
## [1] "# of rows with unique Gene Symbols:"
## [1] 18555

## Warning in dir.create(paste("./", "Collapsing_DEResults_", GSE_ID, sep = "")):
## './Collapsing_DEResults_GSE42264' already exists

## [1] "Dimensions of Fold Change matrix, averaged by gene symbol:"
## [1] 18555      1
## [1] "Output: Collapsing_DEResults_GSE42264"

# GSE44331
Collapsing_DEResults_GSE44331 <- CollapsingDEResults_OneResultPerGene("GSE44331",
                                                                    DE_Results_GSE44331_GoodAnnotation,
                                                                    ComparisonsOfInterest_GSE44331,
                                                                    NamesOfFoldChangeColumns_GSE44331,
                                                                    NamesOfTstatColumns_GSE44331)

```



```

## [1] "Check if the vectors containing FC and Tstat column names contain the same order as the compari.
## [1] "# of rows with unique NCBI IDs:"
## [1] 18557
## [1] "# of rows with unique Gene Symbols:"
## [1] 18557

## Warning in dir.create(paste("./", "Collapsing_DEResults_", GSE_ID, sep = "")):
## './Collapsing_DEResults_GSE44331' already exists

## [1] "Dimensions of Fold Change matrix, averaged by gene symbol:"
## [1] 18557      1
## [1] "Output: Collapsing_DEResults_GSE44331"

# GSE51365
Collapsing_DEResults_GSE51365 <- CollapsingDEResults_OneResultPerGene("GSE51365",
                                                                    DE_Results_GSE51365_GoodAnnotation,
                                                                    ComparisonsOfInterest_GSE51365,
                                                                    NamesOfFoldChangeColumns_GSE51365,
                                                                    NamesOfTstatColumns_GSE51365)

## [1] "Check if the vectors containing FC and Tstat column names contain the same order as the compari.
## [1] "# of rows with unique NCBI IDs:"
## [1] 18561
## [1] "# of rows with unique Gene Symbols:"
## [1] 18561

## Warning in dir.create(paste("./", "Collapsing_DEResults_", GSE_ID, sep = "")):
## './Collapsing_DEResults_GSE51365' already exists

## [1] "Dimensions of Fold Change matrix, averaged by gene symbol:"
## [1] 18561      2
## [1] "Output: Collapsing_DEResults_GSE51365"

# GSE53784
Collapsing_DEResults_GSE53784 <- CollapsingDEResults_OneResultPerGene("GSE53784",
                                                                    DE_Results_GSE53784_GoodAnnotation,
                                                                    ComparisonsOfInterest_GSE53784,
                                                                    NamesOfFoldChangeColumns_GSE53784,
                                                                    NamesOfTstatColumns_GSE53784)

## [1] "Check if the vectors containing FC and Tstat column names contain the same order as the compari.
## [1] "# of rows with unique NCBI IDs:"
## [1] 20248
## [1] "# of rows with unique Gene Symbols:"
## [1] 20248

## Warning in dir.create(paste("./", "Collapsing_DEResults_", GSE_ID, sep = "")):
## './Collapsing_DEResults_GSE53784' already exists

## [1] "Dimensions of Fold Change matrix, averaged by gene symbol:"
## [1] 20248      2
## [1] "Output: Collapsing_DEResults_GSE53784"

```

```

# GSE91074
Collapsing_DEResults_GSE91074 <- CollapsingDEResults_OneResultPerGene("GSE91074",
                                                                    DE_Results_GSE91074_GoodAnnotation,
                                                                    ComparisonsOfInterest_GSE91074,
                                                                    NamesOfFoldChangeColumns_GSE91074,
                                                                    NamesOfTstatColumns_GSE91074)

## [1] "Check if the vectors containing FC and Tstat column names contain the same order as the compari
## [1] "# of rows with unique NCBI IDs:"
## [1] 12644
## [1] "# of rows with unique Gene Symbols:"
## [1] 12644

## Warning in dir.create(paste("./", "Collapsing_DEResults_", GSE_ID, sep = "")):
## './Collapsing_DEResults_GSE91074' already exists

## [1] "Dimensions of Fold Change matrix, averaged by gene symbol:"
## [1] 12644      1
## [1] "Output: Collapsing_DEResults_GSE91074"

```

9) Alignment of DE results from same models (mouse & rat)

Each dataset has DE results from a slightly different list of genes.

Depending on the exact tissue dissected, the sensitivity of the transcriptional profiling platform, the representation on the transcriptional profiling platform (for microarray), and the experimental conditions, the DE results from different datasets will also be in a slightly different order.

We want to align these results so that the DE results from each dataset are columns, with each row representing a different gene.

- GSE30577 - Mouse model
- GSE42264 - Mouse model
- GSE44331 - Mouse model
- GSE51365 - Mouse model
- GSE53784 - Mouse model
- GSE91074 - Mouse model

Since there are only mouse models, we would create one aligning function. If there are both mouse and rat models, we would have to create two aligning functions, one for the mouse and one for the rat.

```

# Create a function to align all mouse DE results from different datasets into
# a single dataframe for Log2FC and SV
AligningMouseDatasets <- function(ListOfMouseDEResults){

  # Create a list to store the log2FC
  Mouse_MetaAnalysis_FoldChange_Dfs <- list()

  # Loop through all mouse DE results

```

```

for(i in c(1:length(ListOfMouseDEResults))){
  # When did the row name become IDs?
  # Extract the Log2FC values from each dataset and put them into separate list
  # The element from the "Collapsing_DEResults_" has the format of row names
  # as Entrez Gene ID and columns containing Log2FC values
  Mouse_MetaAnalysis_FoldChange_Dfs[[i]] <- data.frame(Mouse_EntrezGene.ID = row.names(ListOfMouseDEResults[[i]]),
                                                         ListOfMouseDEResults[[i]][[1]],
                                                         stringsAsFactors = FALSE)
}

# Print the structure of the Log2FC lists
print("Mouse_MetaAnalysis_FoldChange_Dfs:")
print(str(Mouse_MetaAnalysis_FoldChange_Dfs))

# Align the DE results by Entrez Gene ID and turn them into a single dataframe
# join_all can be used for object of list class
Mouse_MetaAnalysis_FoldChanges <-> join_all(Mouse_MetaAnalysis_FoldChange_Dfs,
                                             by = "Mouse_EntrezGene.ID",
                                             type = "full")

# Print the structure of the aligned Log2FC
print("Mouse_MetaAnalysis_FoldChanges:")
print(str(Mouse_MetaAnalysis_FoldChanges))

# Create a list to store the SV
Mouse_MetaAnalysis_SV_Dfs <- list()

# Loop through all mouse DE results
for(i in c(1:length(ListOfMouseDEResults))){
  # Extract the SV values from each dataset and put them into separate list
  # The element from the "Collapsing_DEResults_" has the format of row names
  # as Entrez Gene ID and columns containing SV values
  Mouse_MetaAnalysis_SV_Dfs[[i]] <- data.frame(Mouse_EntrezGene.ID = row.names(ListOfMouseDEResults[[i]]),
                                                         ListOfMouseDEResults[[i]][[4]],
                                                         stringsAsFactors = FALSE)
}

# Print the structure of the SV lists
print("Mouse_MetaAnalysis_SV_Dfs:")
print(str(Mouse_MetaAnalysis_SV_Dfs))

# Align the DE results by Entrez Gene ID and turn them into a single dataframe
# join_all can be used for object of list class
Mouse_MetaAnalysis_SV <-> join_all(Mouse_MetaAnalysis_SV_Dfs,
                                   by = "Mouse_EntrezGene.ID",
                                   type = "full")

# Print the structure of the aligned SV
print("Mouse_MetaAnalysis_SV:")
print(str(Mouse_MetaAnalysis_SV))

# Remove the temporary lists of Log2FC and SV
rm(Mouse_MetaAnalysis_SV_Dfs, Mouse_MetaAnalysis_FoldChange_Dfs)

```

```
}
```

```
# Prepare the list of collapsed mouse DE results
```

```
ListOfMouseDEResults <- list(Collapsing_DEResults_GSE30577,  
                             Collapsing_DEResults_GSE42264,  
                             Collapsing_DEResults_GSE44331,  
                             Collapsing_DEResults_GSE51365,  
                             Collapsing_DEResults_GSE53784,  
                             Collapsing_DEResults_GSE91074)
```

```
# Function use
```

```
AligningMouseDatasets(ListOfMouseDEResults)
```

```
## [1] "Mouse_MetaAnalysis_FoldChange_Dfs:"
```

```
## List of 6
```

```
## $ : 'data.frame': 16125 obs. of 2 variables:  
## ..$ Mouse_EntrezGene.ID : chr [1:16125] "100008567" "100012" "100017" "100019" ...  
## ..$ GSE30577_Rabies_vs_Control: num [1:16125] 2.759 0.333 2.426 -3.133 3.728 ...  
## $ : 'data.frame': 18555 obs. of 2 variables:  
## ..$ Mouse_EntrezGene.ID : chr [1:18555] "100008567" "100009600" "100012" "100017" ...  
## ..$ GSE42264_Measles_vs_Control: num [1:18555] 0.1432 -0.0634 0.0232 0.2999 -0.0068 ...  
## $ : 'data.frame': 18557 obs. of 2 variables:  
## ..$ Mouse_EntrezGene.ID : chr [1:18557] "100008567" "100009600" "100012" "100017" ...  
## ..$ GSE44331_VSV_vs_Control: num [1:18557] -0.1186 -0.0661 0.00045 0.0126 -0.12615 ...  
## $ : 'data.frame': 18561 obs. of 3 variables:  
## ..$ Mouse_EntrezGene.ID : chr [1:18561] "100008567" "100009600" "100012" "100017" ...  
## ..$ GSE51365_MVH68_WT_vs_Control: num [1:18561] -0.027 0.015 -0.1365 0.3838 -0.0438 ...  
## ..$ GSE51365_MVH68_M_vs_Control : num [1:18561] -0.0529 0.0899 -0.1011 0.4863 -0.1134 ...  
## $ : 'data.frame': 20248 obs. of 3 variables:  
## ..$ Mouse_EntrezGene.ID : chr [1:20248] "100008567" "100009600" "100009609" "100009614" ...  
## ..$ GSE53784_WNV_vs_Control: num [1:20248] -0.3712 0.3762 -0.013 0.1307 -0.0234 ...  
## ..$ GSE53784_JEV_vs_Control: num [1:20248] -0.1675 0.1526 0.0452 -0.0839 0.0016 ...  
## $ : 'data.frame': 12644 obs. of 2 variables:  
## ..$ Mouse_EntrezGene.ID : chr [1:12644] "100017" "100019" "100036538" "100036539" ...  
## ..$ GSE91074_VEEV_vs_Control: num [1:12644] 0.7205 -0.1288 -0.1049 -0.0285 0.1944 ...  
## NULL
```

```
## [1] "Mouse_MetaAnalysis_FoldChanges:"
```

```
## 'data.frame': 22542 obs. of 9 variables:
```

```
## $ Mouse_EntrezGene.ID : chr "100008567" "100012" "100017" "100019" ...  
## $ GSE30577_Rabies_vs_Control : num 2.759 0.333 2.426 -3.133 3.728 ...  
## $ GSE42264_Measles_vs_Control : num 0.1432 0.0232 0.2999 -0.0068 0.0324 ...  
## $ GSE44331_VSV_vs_Control : num -0.1186 0.00045 0.0126 -0.12615 0.3555 ...  
## $ GSE51365_MVH68_WT_vs_Control: num -0.027 -0.1365 0.3838 -0.0438 -0.0832 ...  
## $ GSE51365_MVH68_M_vs_Control : num -0.0529 -0.1011 0.4863 -0.1134 0.0224 ...  
## $ GSE53784_WNV_vs_Control : num -0.3712 -0.0234 0.1029 0.2022 3.5291 ...  
## $ GSE53784_JEV_vs_Control : num -0.1675 0.0016 -0.2338 0.2673 2.316 ...  
## $ GSE91074_VEEV_vs_Control : num NA NA 0.721 -0.129 NA ...
```

```
## NULL
```

```
## [1] "Mouse_MetaAnalysis_SV_Dfs:"
```

```
## List of 6
```

```
## $ : 'data.frame': 16125 obs. of 2 variables:  
## ..$ Mouse_EntrezGene.ID : chr [1:16125] "100008567" "100012" "100017" "100019" ...  
## ..$ GSE30577_Rabies_vs_Control: num [1:16125] 1.458 1.986 0.378 0.953 0.547 ...
```

```
## $ : 'data.frame': 18555 obs. of 2 variables:
## ..$ Mouse_EntrezGene.ID : chr [1:18555] "100008567" "100009600" "100012" "100017" ...
## ..$ GSE42264_Measles_vs_Control: num [1:18555] 0.00544 0.00205 0.00381 0.00488 0.00255 ...
## $ : 'data.frame': 18557 obs. of 2 variables:
## ..$ Mouse_EntrezGene.ID : chr [1:18557] "100008567" "100009600" "100012" "100017" ...
## ..$ GSE44331_VSV_vs_Control: num [1:18557] 0.01038 0.00728 0.00602 0.01243 0.00568 ...
## $ : 'data.frame': 18561 obs. of 3 variables:
## ..$ Mouse_EntrezGene.ID : chr [1:18561] "100008567" "100009600" "100012" "100017" ...
## ..$ GSE51365_MVH68_WT_vs_Control: num [1:18561] 0.00531 0.00645 0.00989 0.06474 0.00793 ...
## ..$ GSE51365_MVH68_M_vs_Control : num [1:18561] 0.00532 0.00644 0.00987 0.06476 0.00795 ...
## $ : 'data.frame': 20248 obs. of 3 variables:
## ..$ Mouse_EntrezGene.ID : chr [1:20248] "100008567" "100009600" "100009609" "100009614" ...
## ..$ GSE53784_WNV_vs_Control: num [1:20248] 0.00481 0.00939 0.00889 0.0103 0.0035 ...
## ..$ GSE53784_JEV_vs_Control: num [1:20248] 0.00481 0.00939 0.00888 0.01031 0.00362 ...
## $ : 'data.frame': 12644 obs. of 2 variables:
## ..$ Mouse_EntrezGene.ID : chr [1:12644] "100017" "100019" "100036538" "100036539" ...
## ..$ GSE91074_VEEV_vs_Control: num [1:12644] 0.01898 0.00278 0.00226 0.00209 0.00528 ...
## NULL
## [1] "Mouse_MetaAnalysis_SV:"
## 'data.frame': 22542 obs. of 9 variables:
## $ Mouse_EntrezGene.ID : chr "100008567" "100012" "100017" "100019" ...
## $ GSE30577_Rabies_vs_Control : num 1.458 1.986 0.378 0.953 0.547 ...
## $ GSE42264_Measles_vs_Control : num 0.00544 0.00381 0.00488 0.00255 0.00586 ...
## $ GSE44331_VSV_vs_Control : num 0.01038 0.00602 0.01243 0.00568 0.01053 ...
## $ GSE51365_MVH68_WT_vs_Control: num 0.00531 0.00989 0.06474 0.00793 0.0779 ...
## $ GSE51365_MVH68_M_vs_Control : num 0.00532 0.00987 0.06476 0.00795 0.07782 ...
## $ GSE53784_WNV_vs_Control : num 0.00481 0.0035 0.00595 0.00341 0.02618 ...
## $ GSE53784_JEV_vs_Control : num 0.00481 0.00362 0.00595 0.00341 0.02618 ...
## $ GSE91074_VEEV_vs_Control : num NA NA 0.01898 0.00278 NA ...
## NULL
```

10) Alignment of DE results from different models (mouse & rat)

Concept of *Gene Orthologs*:

- *Homology* refers to biological features including genes and their products that are descended from a feature present in a common ancestor.
- *Homologous gene* become separated in evolution in 2 different ways:

1. Separation of 2 populations with the ancestral gene into 2 species

Gene separated by *speciation* are called *orthologs*.

2. Duplication of the ancestral gene within a lineage

Gene separated by *gene duplication* are called *paralogs*.

Ref: https://www.nlm.nih.gov/ncbi/workshops/2023-08_BLAST_evol/ortho_para.html

We have the ortholog database that we downloaded from Jackson Lab on April 25, 2024. This database was trimmed and formatted using “FormattingRatMouseOrthologDatabase_20240425.R”

```

# Import the trimmed and formatted ortholog database
MouseVsRat_NCBI_Entrez <- read.csv("MouseVsRat_NCBI_Entrez_JacksonLab_20240425.csv",
                                   header = TRUE,
                                   stringsAsFactors = FALSE,
                                   row.names = 1,
                                   colClasses = c("character", "character", "character"))

# Join the ortholog database with mouse Log2FC
Mouse_MetaAnalysis_FoldChanges_wOrthologs <- join(MouseVsRat_NCBI_Entrez,
                                                  Mouse_MetaAnalysis_FoldChanges,
                                                  by = "Mouse_EntrezGene.ID",
                                                  type = "full")

# Join the ortholog database with mouse SV
Mouse_MetaAnalysis_SV_wOrthologs <- join(MouseVsRat_NCBI_Entrez,
                                          Mouse_MetaAnalysis_SV,
                                          by = "Mouse_EntrezGene.ID",
                                          type = "full")

# Check the structure of the new mouse Log2FC with ortholog info
str(Mouse_MetaAnalysis_FoldChanges_wOrthologs)

```

```

## 'data.frame': 22789 obs. of 11 variables:
## $ Rat_EntrezGene.ID : chr "114087" "191569" "246307" "65041" ...
## $ Mouse_EntrezGene.ID : chr "23825" "18585" "66514" "20480" ...
## $ MouseVsRat_EntrezGene.ID : chr "23825_114087" "18585_191569" "66514_246307" "20480_65041" ...
## $ GSE30577_Rabies_vs_Control : num -0.521 1.872 NA 0.134 0.822 ...
## $ GSE42264_Measles_vs_Control : num 0.0928 0.0906 0.0443 -0.1149 0.0472 ...
## $ GSE44331_VSV_vs_Control : num -0.00553 0.0339 -0.0275 -0.19225 -0.1228 ...
## $ GSE51365_MVH68_WT_vs_Control: num -0.011 0.1377 0.0275 0.0508 -0.0837 ...
## $ GSE51365_MVH68_M_vs_Control : num 0.00693 0.0891 0.03023 0.0812 -0.095 ...
## $ GSE53784_WNV_vs_Control : num -0.0736 -0.4003 -1.0309 -0.2629 0.8745 ...
## $ GSE53784_JEV_vs_Control : num 0.0783 0.2109 -0.996 -0.0471 0.4395 ...
## $ GSE91074_VEEV_vs_Control : num -0.1555 0.0199 -0.2302 -0.2588 0.0919 ...

```

```

# Check the structure of the new mouse SV with ortholog info
str(Mouse_MetaAnalysis_SV_wOrthologs)

```

```

## 'data.frame': 22789 obs. of 11 variables:
## $ Rat_EntrezGene.ID : chr "114087" "191569" "246307" "65041" ...
## $ Mouse_EntrezGene.ID : chr "23825" "18585" "66514" "20480" ...
## $ MouseVsRat_EntrezGene.ID : chr "23825_114087" "18585_191569" "66514_246307" "20480_65041" ...
## $ GSE30577_Rabies_vs_Control : num 0.781 3.343 NA 1.338 1.128 ...
## $ GSE42264_Measles_vs_Control : num 0.00148 0.00141 0.0018 0.00159 0.00108 ...
## $ GSE44331_VSV_vs_Control : num 0.01018 0.00379 0.00519 0.00888 0.00784 ...
## $ GSE51365_MVH68_WT_vs_Control: num 0.00552 0.00613 0.00682 0.00798 0.00881 ...
## $ GSE51365_MVH68_M_vs_Control : num 0.00545 0.00612 0.00682 0.008 0.00882 ...
## $ GSE53784_WNV_vs_Control : num 0.00586 0.00632 0.00421 0.00281 0.00332 ...
## $ GSE53784_JEV_vs_Control : num 0.00586 0.00632 0.00421 0.00282 0.00332 ...
## $ GSE91074_VEEV_vs_Control : num 0.00239 0.02259 0.00431 0.00414 0.00198 ...

```

```
# If there are rat datasets, we want to join our mouse Log2FC and SV results to  
# the rat Log2FC and SV results using the ortholog info
```

```
# Join the Log2FC of the mouse datasets and rat datasets  
MetaAnalysis_FoldChanges <- join(Mouse_MetaAnalysis_FoldChanges_wOrthologs,  
                                Rat_MetaAnalysis_FoldChanges,  
                                by = "Rat_EntrezGene.ID",  
                                type = "full")
```

```
# Check the structure of the joined mouse and rat Log2FC with ortholog info  
str(MetaAnalysis_FoldChanges)
```

```
# Join the SV of the mouse datasets and rat datasets  
MetaAnalysis_SV <- join(Mouse_MetaAnalysis_SV_wOrthologs,  
                        Rat_MetaAnalysis_FoldChanges,  
                        by = "Rat_EntrezGene.ID",  
                        type = "full")
```

```
# Check the structure of the joined mouse and rat SV with ortholog info  
str(MetaAnalysis_SV)
```

```
# If there are no rat dataset, we rename the dataframes so the pipeline code works  
MetaAnalysis_FoldChanges <- Mouse_MetaAnalysis_FoldChanges_wOrthologs  
MetaAnalysis_SV <- Mouse_MetaAnalysis_SV_wOrthologs
```

```
# Rename Mouse_Rat Entrez annotation for Log2FC dataframe  
MetaAnalysis_FoldChanges$MouseVsRat_EntrezGene.ID <- paste(MetaAnalysis_FoldChanges$Mouse_EntrezGene.ID,  
                                                           MetaAnalysis_FoldChanges$Rat_EntrezGene.ID,  
                                                           sep="_")
```

```
# Rename Mouse_Rat Entrez annotation for SV dataframe  
MetaAnalysis_SV$MouseVsRat_EntrezGene.ID <- paste(MetaAnalysis_SV$Mouse_EntrezGene.ID,  
                                                  MetaAnalysis_SV$Rat_EntrezGene.ID,  
                                                  sep="_")
```

```
# Check the structure of the Log2FC dataframe  
str(MetaAnalysis_FoldChanges)
```

```
## 'data.frame':    22789 obs. of  11 variables:  
## $ Rat_EntrezGene.ID      : chr  "114087" "191569" "246307" "65041" ...  
## $ Mouse_EntrezGene.ID    : chr  "23825" "18585" "66514" "20480" ...  
## $ MouseVsRat_EntrezGene.ID : chr  "23825_114087" "18585_191569" "66514_246307" "20480_65041" ...  
## $ GSE30577_Rabies_vs_Control : num  -0.521 1.872 NA 0.134 0.822 ...  
## $ GSE42264_Measles_vs_Control : num  0.0928 0.0906 0.0443 -0.1149 0.0472 ...  
## $ GSE44331_VSV_vs_Control : num  -0.00553 0.0339 -0.0275 -0.19225 -0.1228 ...  
## $ GSE51365_MVH68_WT_vs_Control: num  -0.011 0.1377 0.0275 0.0508 -0.0837 ...  
## $ GSE51365_MVH68_M_vs_Control : num  0.00693 0.0891 0.03023 0.0812 -0.095 ...  
## $ GSE53784_WNV_vs_Control : num  -0.0736 -0.4003 -1.0309 -0.2629 0.8745 ...  
## $ GSE53784_JEV_vs_Control : num  0.0783 0.2109 -0.996 -0.0471 0.4395 ...  
## $ GSE91074_VEEV_vs_Control : num  -0.1555 0.0199 -0.2302 -0.2588 0.0919 ...
```

```
# Check the structure of the SV dataframe
str(MetaAnalysis_SV)
```

```
## 'data.frame': 22789 obs. of 11 variables:
## $ Rat_EntrezGene.ID : chr "114087" "191569" "246307" "65041" ...
## $ Mouse_EntrezGene.ID : chr "23825" "18585" "66514" "20480" ...
## $ MouseVsRat_EntrezGene.ID : chr "23825_114087" "18585_191569" "66514_246307" "20480_65041" ...
## $ GSE30577_Rabies_vs_Control : num 0.781 3.343 NA 1.338 1.128 ...
## $ GSE42264_Measles_vs_Control : num 0.00148 0.00141 0.0018 0.00159 0.00108 ...
## $ GSE44331_VSV_vs_Control : num 0.01018 0.00379 0.00519 0.00888 0.00784 ...
## $ GSE51365_MVH68_WT_vs_Control : num 0.00552 0.00613 0.00682 0.00798 0.00881 ...
## $ GSE51365_MVH68_M_vs_Control : num 0.00545 0.00612 0.00682 0.008 0.00882 ...
## $ GSE53784_WNV_vs_Control : num 0.00586 0.00632 0.00421 0.00281 0.00332 ...
## $ GSE53784_JEV_vs_Control : num 0.00586 0.00632 0.00421 0.00282 0.00332 ...
## $ GSE91074_VEEV_vs_Control : num 0.00239 0.02259 0.00431 0.00414 0.00198 ...
```

11) Comparison of Log2FC across datasets

```
# Check the column names in the MetaAnalysis dataframes
colnames(MetaAnalysis_FoldChanges)
```

```
## [1] "Rat_EntrezGene.ID" "Mouse_EntrezGene.ID"
## [3] "MouseVsRat_EntrezGene.ID" "GSE30577_Rabies_vs_Control"
## [5] "GSE42264_Measles_vs_Control" "GSE44331_VSV_vs_Control"
## [7] "GSE51365_MVH68_WT_vs_Control" "GSE51365_MVH68_M_vs_Control"
## [9] "GSE53784_WNV_vs_Control" "GSE53784_JEV_vs_Control"
## [11] "GSE91074_VEEV_vs_Control"
```

There are different ways to plot relationships across datasets:

1. Rank-rank hypergeometric overlap plots
2. Hierarchically clustered heatmaps

We can generally compare the DE results associated with different contrasts using a scatter plot and correlation analysis.

```
# Create a correlation matrix for Log2FC across datasets
# "pairwise.complete.obs" ignore any rows of DE results that do not have Log2FC
# for one of our columns
MetaAnalysis_CorMatrix_FoldChanges <- cor(as.matrix(MetaAnalysis_FoldChanges[, -c(1:3)]),
                                           use = "pairwise.complete.obs",
                                           method = "spearman")

# Check the correlation matrix
MetaAnalysis_CorMatrix_FoldChanges
```

```
##                                GSE30577_Rabies_vs_Control
## GSE30577_Rabies_vs_Control      1.00000000
## GSE42264_Measles_vs_Control      0.14675164
## GSE44331_VSV_vs_Control          0.19172649
```


##	GSE51365_MVH68_WT_vs_Control	0.01774524	
##	GSE51365_MVH68_M_vs_Control	-0.01817159	
##	GSE53784_WNV_vs_Control	0.35176909	
##	GSE53784_JEV_vs_Control	0.33712002	
##	GSE91074_VEEV_vs_Control	0.46330095	
##	GSE42264_Measles_vs_Control		
##	GSE30577_Rabies_vs_Control	0.14675164	
##	GSE42264_Measles_vs_Control	1.00000000	
##	GSE44331_VSV_vs_Control	0.16058484	
##	GSE51365_MVH68_WT_vs_Control	-0.00418835	
##	GSE51365_MVH68_M_vs_Control	0.02661880	
##	GSE53784_WNV_vs_Control	0.24596626	
##	GSE53784_JEV_vs_Control	0.24290161	
##	GSE91074_VEEV_vs_Control	0.30976676	
##	GSE44331_VSV_vs_Control		
##	GSE30577_Rabies_vs_Control	0.1917265	
##	GSE42264_Measles_vs_Control	0.1605848	
##	GSE44331_VSV_vs_Control	1.0000000	
##	GSE51365_MVH68_WT_vs_Control	0.1150817	
##	GSE51365_MVH68_M_vs_Control	0.1133153	
##	GSE53784_WNV_vs_Control	0.1612147	
##	GSE53784_JEV_vs_Control	0.1862596	
##	GSE91074_VEEV_vs_Control	0.2502848	
##	GSE51365_MVH68_WT_vs_Control		
##	GSE30577_Rabies_vs_Control	0.01774524	
##	GSE42264_Measles_vs_Control	-0.00418835	
##	GSE44331_VSV_vs_Control	0.11508169	
##	GSE51365_MVH68_WT_vs_Control	1.00000000	
##	GSE51365_MVH68_M_vs_Control	0.52913023	
##	GSE53784_WNV_vs_Control	-0.03381511	
##	GSE53784_JEV_vs_Control	-0.02096606	
##	GSE91074_VEEV_vs_Control	0.01657012	
##	GSE51365_MVH68_M_vs_Control		
##	GSE30577_Rabies_vs_Control	-0.018171586	
##	GSE42264_Measles_vs_Control	0.026618798	
##	GSE44331_VSV_vs_Control	0.113315346	
##	GSE51365_MVH68_WT_vs_Control	0.529130232	
##	GSE51365_MVH68_M_vs_Control	1.000000000	
##	GSE53784_WNV_vs_Control	-0.059384072	
##	GSE53784_JEV_vs_Control	-0.054283220	
##	GSE91074_VEEV_vs_Control	-0.009155205	
##	GSE53784_WNV_vs_Control	GSE53784_JEV_vs_Control	
##	GSE30577_Rabies_vs_Control	0.35176909	0.33712002
##	GSE42264_Measles_vs_Control	0.24596626	0.24290161
##	GSE44331_VSV_vs_Control	0.16121467	0.18625956
##	GSE51365_MVH68_WT_vs_Control	-0.03381511	-0.02096606
##	GSE51365_MVH68_M_vs_Control	-0.05938407	-0.05428322
##	GSE53784_WNV_vs_Control	1.00000000	0.77150650
##	GSE53784_JEV_vs_Control	0.77150650	1.00000000
##	GSE91074_VEEV_vs_Control	0.63233330	0.61308916
##	GSE91074_VEEV_vs_Control		
##	GSE30577_Rabies_vs_Control	0.463300945	
##	GSE42264_Measles_vs_Control	0.309766756	
##	GSE44331_VSV_vs_Control	0.250284764	

```
## GSE51365_MVH68_WT_vs_Control      0.016570117
## GSE51365_MVH68_M_vs_Control      -0.009155205
## GSE53784_WNV_vs_Control          0.632333296
## GSE53784_JEV_vs_Control          0.613089157
## GSE91074_VEEV_vs_Control          1.000000000
```

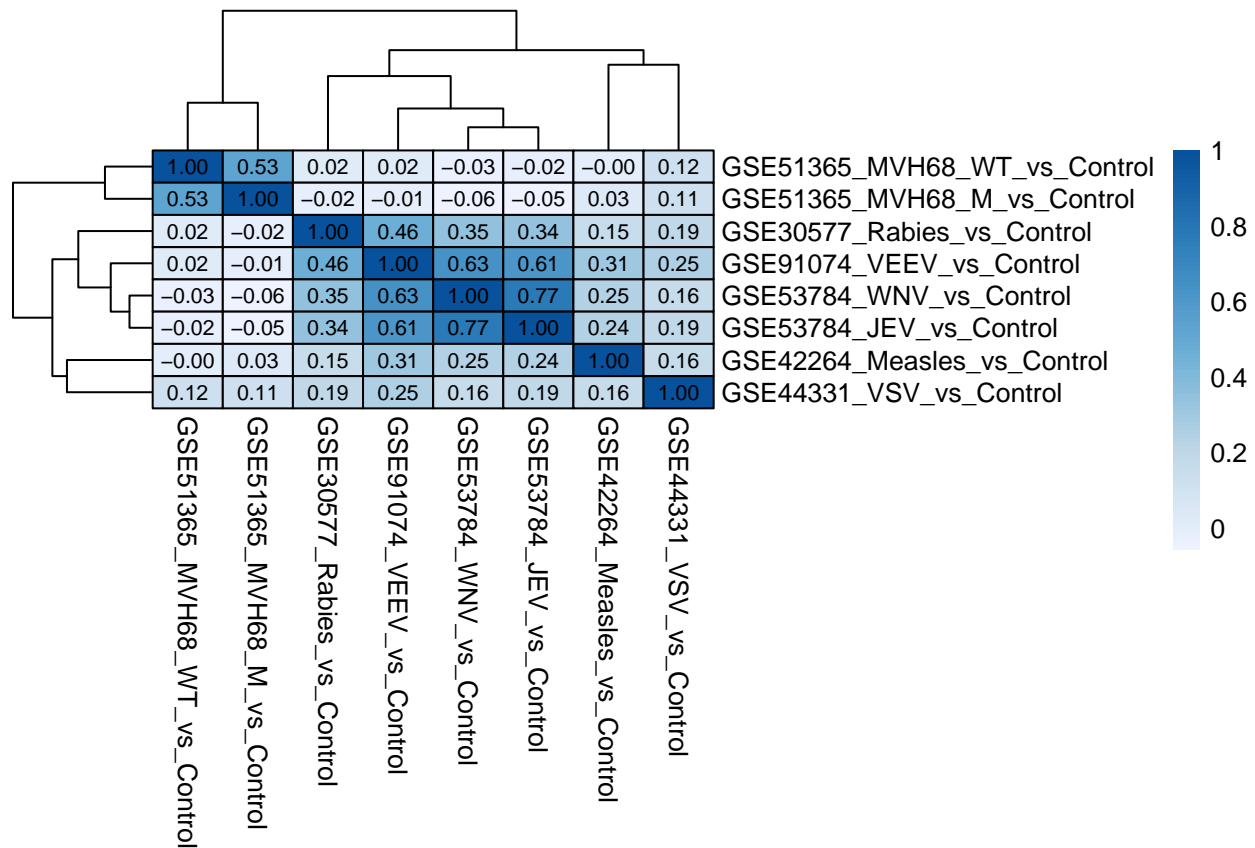
Each cell includes the correlation coefficient reflecting the similarity of the effect sizes for the comparison of between the contrast in the row and the according contrast in the column.

Correlation coefficient ranges between -1 to 1, with -1 being a perfect negative correlation and +1 being a perfect positive correlation

Illustrate the correlation matrix with hierarchically clustered heatmap

Blues Palette

```
pheatmap(MetaAnalysis_CorMatrix_FoldChanges,
  display_numbers = TRUE,
  fontsize = 10,
  number_format = "%.2f",
  color = colorRampPalette(brewer.pal(5, "Blues"))(100),
  border_color = "black",
  number_color = "black")
```



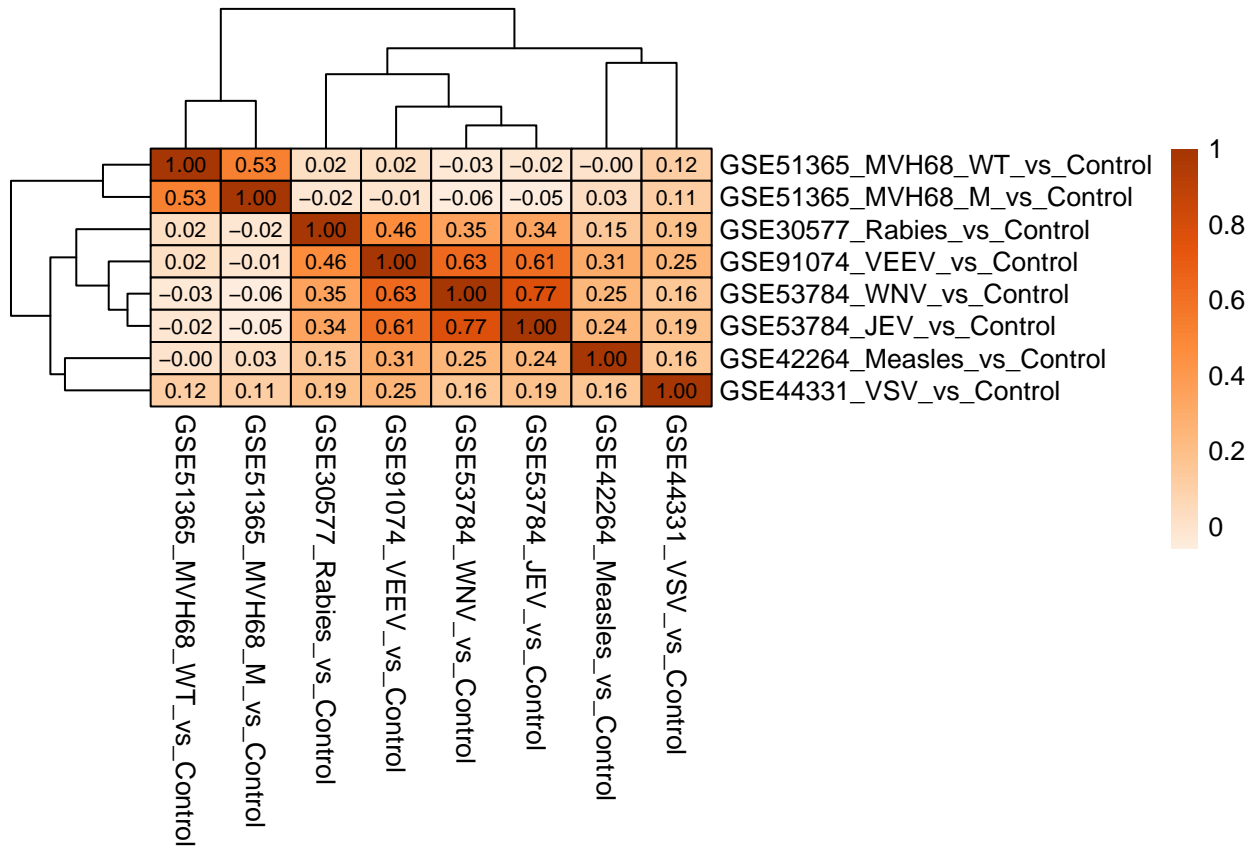
Oranges Palette

```
pheatmap(MetaAnalysis_CorMatrix_FoldChanges,
  display_numbers = TRUE,
```

```

fontsize = 10,
number_format = "%.2f",
color = colorRampPalette(brewer.pal(5, "Oranges"))(100),
border_color = "black",
number_color = "black")

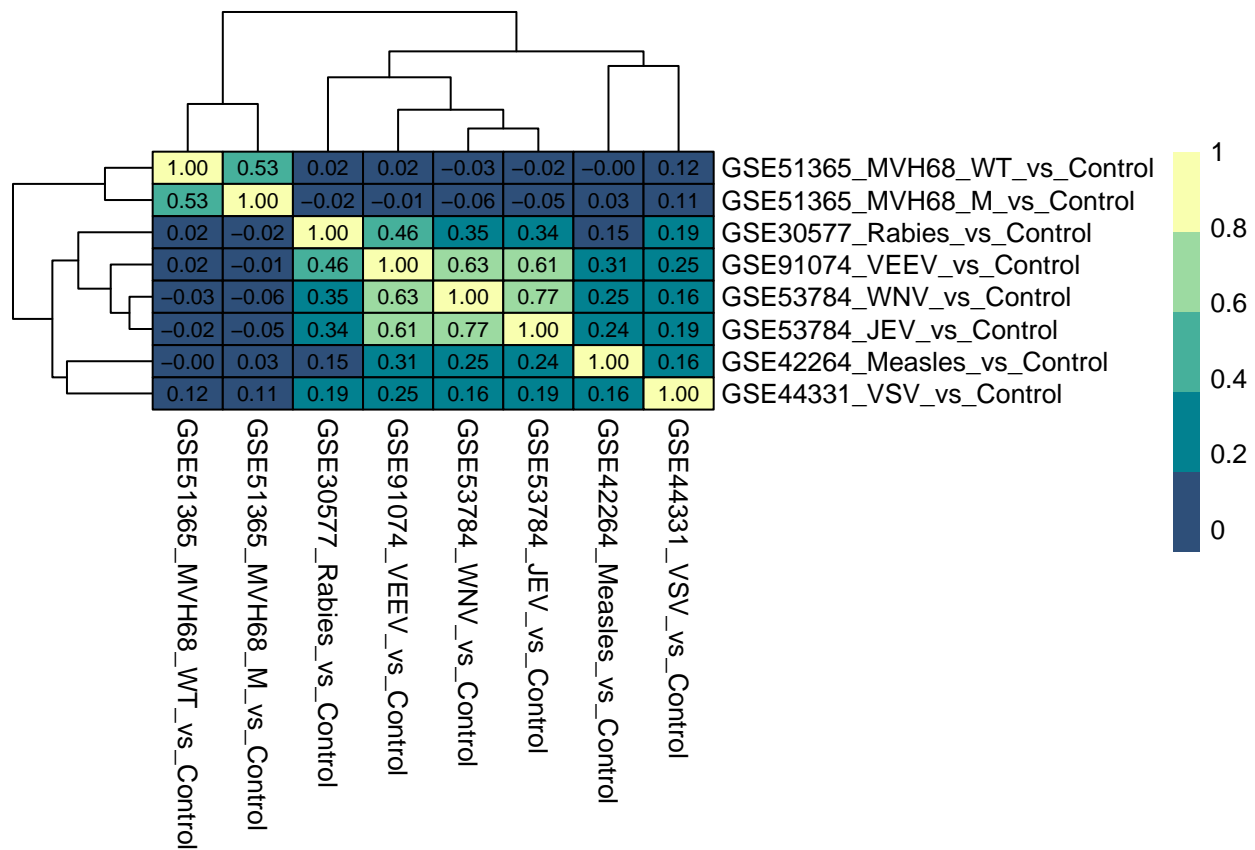
```



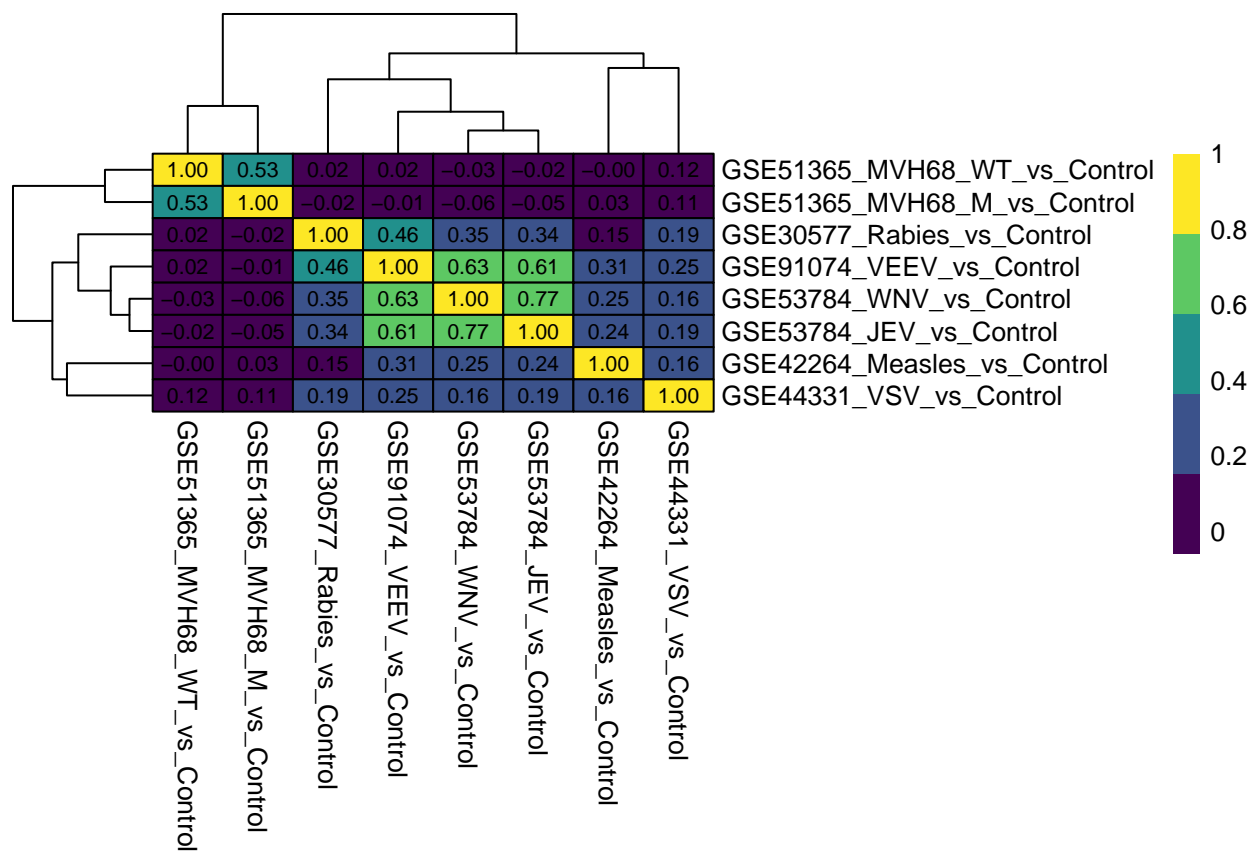
```

# Blues-Yellows Palette
pheatmap(MetaAnalysis_CorMatrix_FoldChanges,
display_numbers = TRUE,
fontsize = 10,
number_format = "%.2f",
color = hcl.colors(5, "BluYl"),
border_color = "black",
number_color = "black")

```



```
# Viridis Palette
pheatmap(MetaAnalysis_CorMatrix_FoldChanges,
  display_numbers = TRUE,
  fontsize = 10,
  number_format = "%.2f",
  color = viridis(5),
  border_color = "black",
  number_color = "black")
```



In this heatmap, white/yellow indicates a more positive correlation
 # The groups are placed in order by similarity, as determined by hierarchical clustering
 # The lines ("tree branches") on the left and top illustrate that similarity (clustering)
 # using a "dendrogram"