Integrative Expression matrix from GSE24752, GSE28345, and GSE28360 for transcriptomic analysis of Hypertension

Loranda_Calderon

2025-06-01

```r
library(GEOquery)
```

```
## Loading required package: Biobase

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

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

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

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

## Setting options('download.file.method.GEOquery'='auto')

## Setting options('GEOquery.inmemory.gpl'=FALSE)
```

```r
library(affy)
library(limma)
```

```
##
## Attaching package: 'limma'
## The following object is masked from ’package:BiocGenerics’:
##
##     plotMA
```

```
library(sva)
```

```
## Loading required package: mgcv
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.9-1. For overview type ’help("mgcv-package")’.
```

```
## Loading required package: genefilter
```

```
## Loading required package: BiocParallel
```

```
library(WGCNA)
```

```
## Loading required package: dynamicTreeCut
```

```
## Loading required package: fastcluster
```

```
##
## Attaching package: ’fastcluster’
```

```
## The following object is masked from ’package:stats’:
##
##     hclust
```

```
##
```

```
##
## Attaching package: ’WGCNA’
```

```
## The following object is masked from ’package:stats’:
##
##     cor
```

```
library(ensembldb)
```

```
## Loading required package: GenomicRanges
```

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

```
## Loading required package: S4Vectors
```

```
##
## Attaching package: ’S4Vectors’
## The following object is masked from 'package:utils':
##
##     findMatches

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

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:nlme':
##
##     collapse

## Loading required package: GenomeInfoDb

## Loading required package: GenomicFeatures

## Loading required package: AnnotationDbi

## Loading required package: AnnotationFilter

##
## Attaching package: 'ensembldb'

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

```
library(biomaRt)
library(arrayQualityMetrics)
```

```
# Download GSE24752 data (HTN)
gse_HTN <- getGEO("GSE24752", GSEMatrix =TRUE,getGPL=FALSE)
```

```
## Found 1 file(s)

## GSE24752_series_matrix.txt.gz
```

```
datMeta_HTN <- pData(gse_HTN[[1]])
rownames(datMeta_HTN) <- datMeta_HTN$geo_accession
```

```
# Read GSE24752 data
setwd("/Users/lorandacalderonzamora/GSE24752/")
data.affy_HTN <- ReadAffy(celfile.path = "./")
datExpr_HTN <- exprs(data.affy_HTN)
```

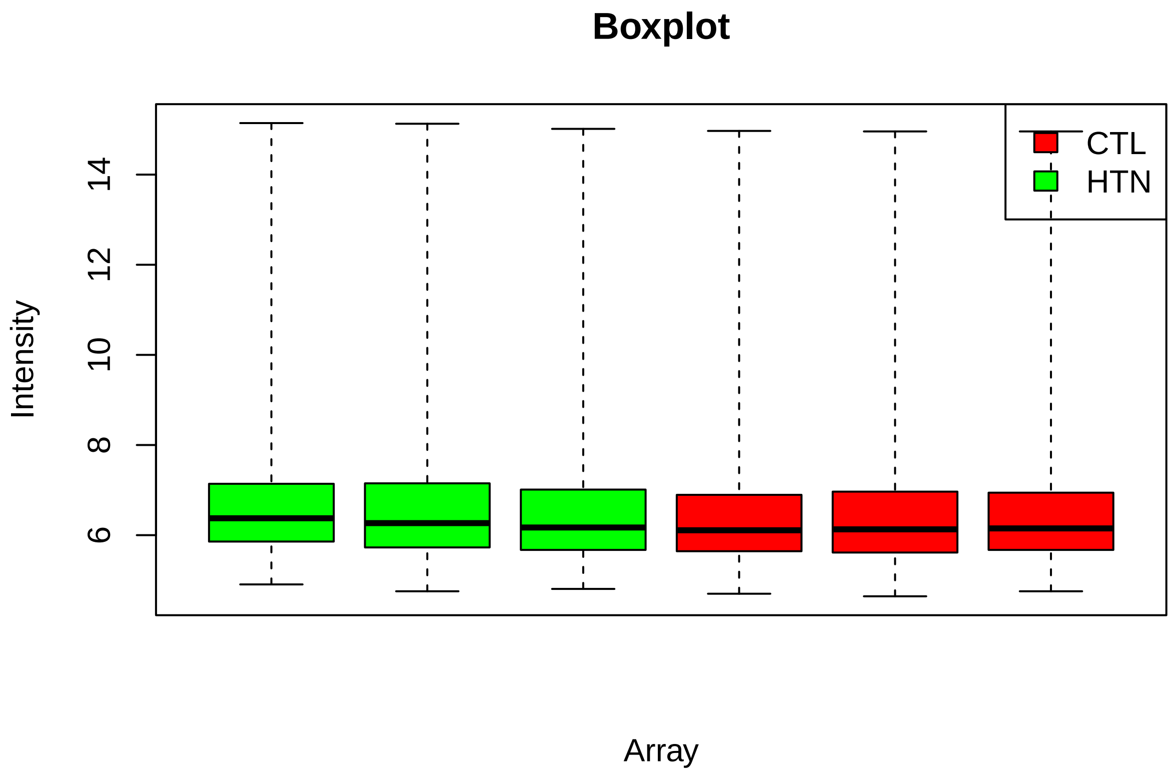
```r
# Align datMeta_HTN and datExpr_HTN by sample identifiers
GSM_HTN <- rownames(pData(data.affy_HTN))
GSM_HTN <- substr(GSM_HTN, 1, 9)
idx_HTN <- match(GSM_HTN, datMeta_HTN$geo_accession)
datMeta_HTN <- datMeta_HTN[idx_HTN, ]
colnames(datExpr_HTN) <- rownames(datMeta_HTN)
```

```r
# Cleaning and formatting of GSE24752 metadata
datMeta_HTN <- datMeta_HTN[, -c(3:7, 14:36)]
colnames(datMeta_HTN)[2] <- "Dx"
datMeta_HTN$Dx[rownames(datMeta_HTN) %in% c("GSM609528", "GSM609529", "GSM609530")] <- "CTL"
datMeta_HTN$Dx[rownames(datMeta_HTN) %in% c("GSM609525", "GSM609526", "GSM609527")] <- "HTN"
datMeta_HTN$Dx <- as.factor(datMeta_HTN$Dx)
```

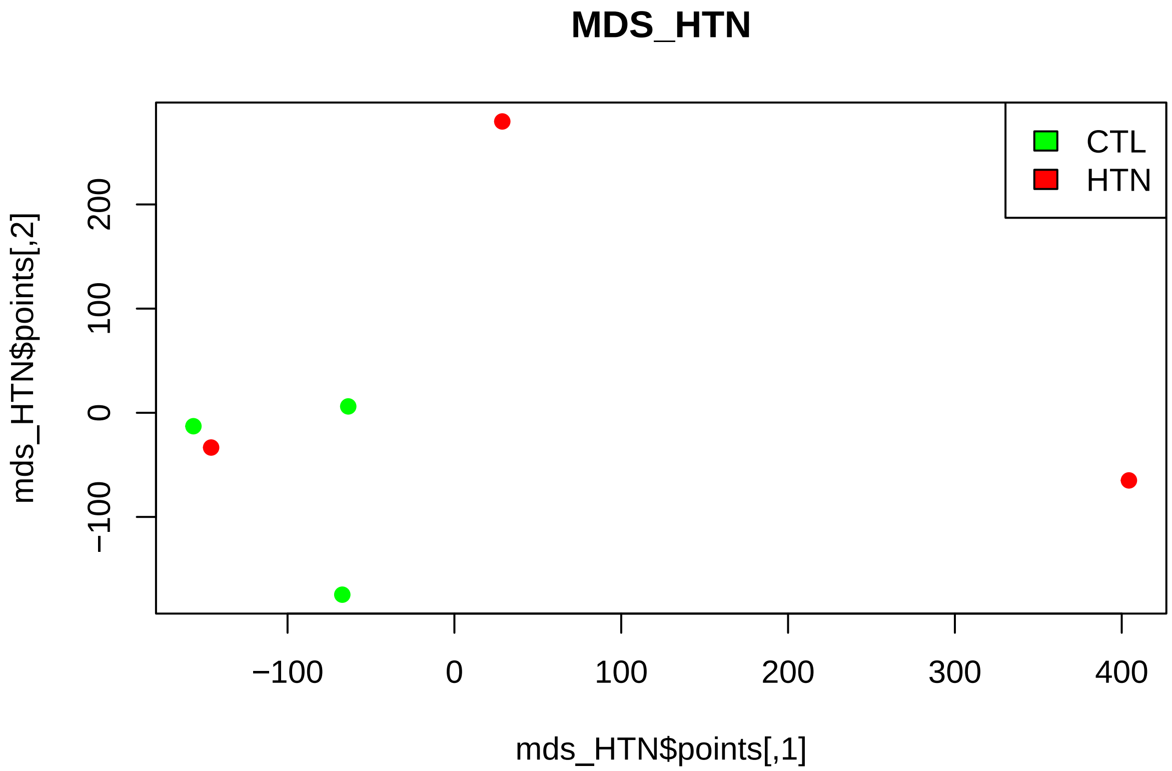
```r
# Preprocessing and quality assessment of GSE24752 raw expression data
datExpr_HTN <- log2(datExpr_HTN)
dim(datExpr_HTN)
```

```
## [1] 1354896       6
```

```r
# Exploratory visualization of GSE24752 raw data
boxplot(datExpr_HTN,range=0, col=c('red', 'green')[as.numeric(datMeta_HTN$Dx)], xaxt='n', xlab = "Array
legend("topright",legend = levels(datMeta_HTN$Dx),fill = c('red', 'green')[as.numeric(as.factor(levels(
```



```
mds_HTN = cmdscale(dist(t(datExpr_HTN)),eig=TRUE)
plot(mds_HTN$points,col=c('green', 'red')[as.numeric(datMeta_HTN$Dx)],pch=19,main="MDS_HTN")
legend("topright",legend = levels(datMeta_HTN$Dx),fill =c('green', 'red')[as.numeric(as.factor(levels(da
```

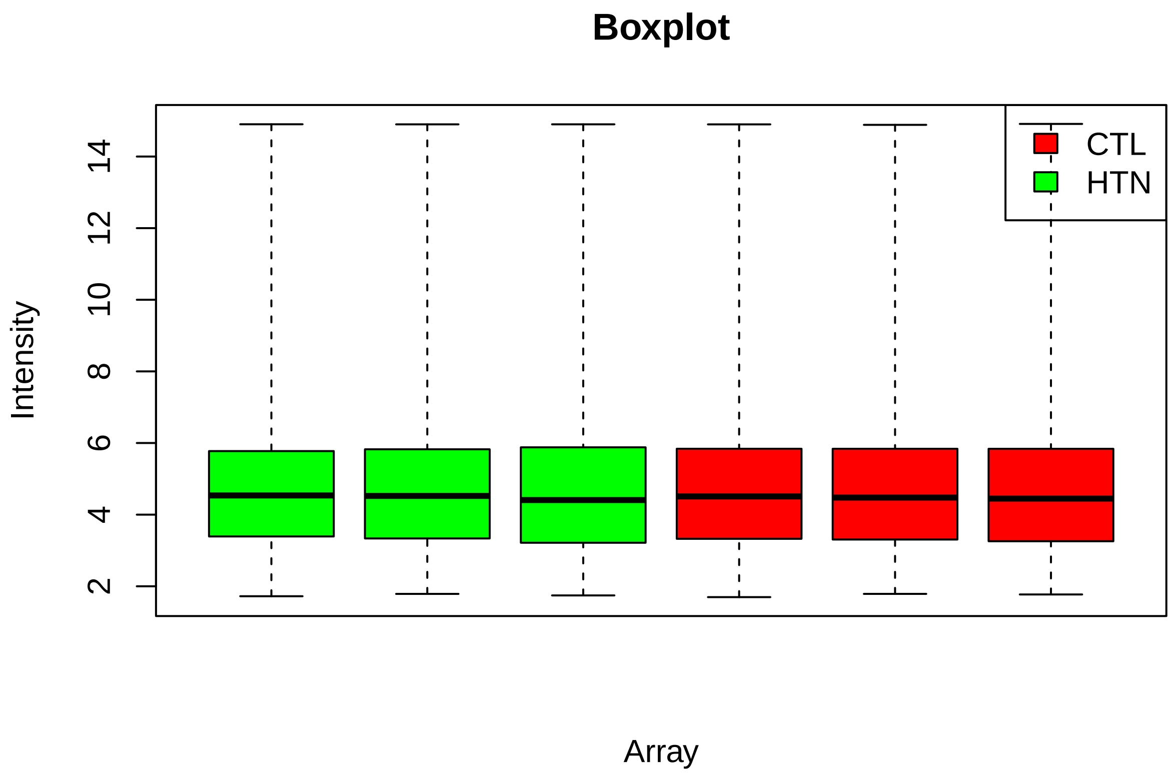


```
# Normalization using RMA
datExpr_HTN <- rma(data.affy_HTN, background=T, normalize=T, verbose=T)
```

```
## Warning: replacing previous import 'AnnotationDbi::tail' by 'utils::tail' when
## loading 'hgu133plus2cdf'

## Warning: replacing previous import 'AnnotationDbi::head' by 'utils::head' when
## loading 'hgu133plus2cdf'

##

## Background correcting
## Normalizing
## Calculating Expression
```

```
datExpr_HTN <- exprs(datExpr_HTN)
```

```
# Exploratory visualization of GSE24752 normalized data
boxplot(datExpr_HTN,range=0, col=c('red', 'green')[as.numeric(datMeta_HTN$Dx)], xaxt='n', xlab = "Array
legend("topright",legend = levels(datMeta_HTN$Dx),fill = c('red', 'green')[as.numeric(as.factor(levels(d
```



```
# QC analysis with arrayQualityMetrics
datMeta_proc_HTN <- new("AnnotatedDataFrame", data = datMeta_HTN)
colnames(datExpr_HTN) <- rownames(datMeta_HTN)
eset_HTN <- new("ExpressionSet", exprs = datExpr_HTN, phenoData = datMeta_proc_HTN)

arrayQualityMetrics(expressionset = eset_HTN,
                    outdir = "/Users/lorandacalderonzamora/Downloads/QC_GSE24752_Report",
                    force = TRUE,
                    do.logtransform = FALSE)
```

```
## The directory '/Users/lorandacalderonzamora/Downloads/QC_GSE24752_Report' has been created.
```

```
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute
## name(s): subscripts, group.number, group.value
```

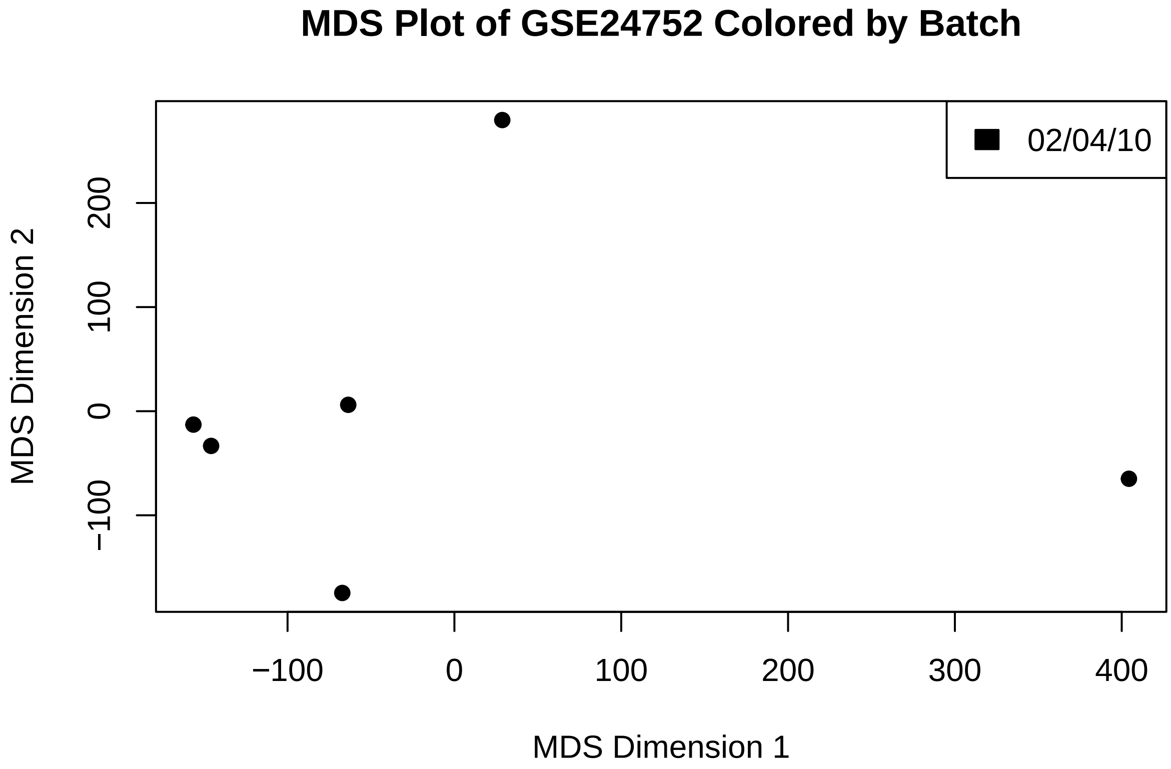
```
## (loaded the KernSmooth namespace)
```

```
# Extract ScanDate from GSE24752 for batch effect correction
batch_HTN <- protocolData(data.affy_HTN)$ScanDate
batch_HTN <- substr(batch_HTN,1,8)
batch_HTN <- as.factor(batch_HTN)
table(batch_HTN)
```

```
## batch_HTN
## 02/04/10
##        6
```

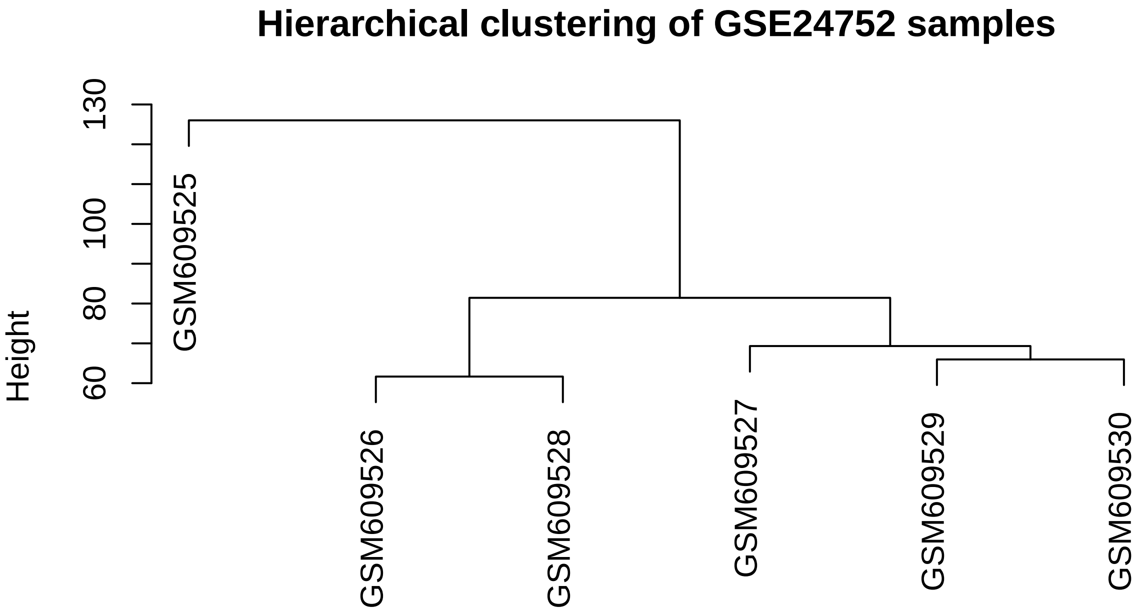
```
datMeta_HTN$Batch <- batch_HTN
```

```
# Visualization of ScanDate metadata from GSE24752 to identify potential batch effects
plot(mds_HTN$points,col = as.numeric(datMeta_HTN$Batch),pch=19,main="MDS Plot of GSE24752 Colored by Ba
legend("topright",legend = levels(datMeta_HTN$Batch),fill = as.numeric(as.factor(levels(datMeta_HTN$Batc
```

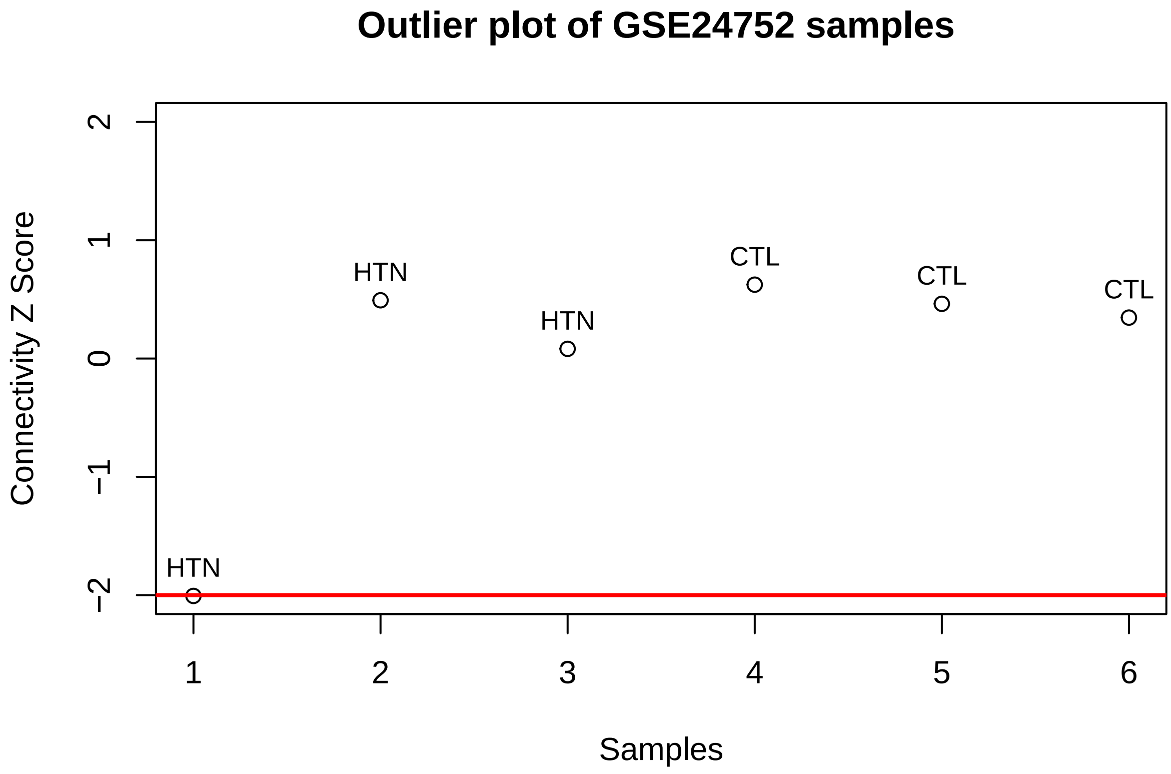


```
# Create ExpressionSet object after Batch effect assessment
datMeta_HTN$Batch <- batch_HTN
datMeta_proc_HTN <- new("AnnotatedDataFrame", data = datMeta_HTN)
colnames(datExpr_HTN) <- rownames(datMeta_HTN)
datAll_HTN <- new("ExpressionSet", exprs = datExpr_HTN, phenoData = datMeta_proc_HTN)
# No singular batch was detected in the GSE24752 dataset.
# Therefore, batch correction with ComBat is technically feasible.
# However, as no evident batch effect was observed in exploratory analyses (MDS),
# ComBat was not applied, and no batch removal was necessary.
```

```
# Sample Clustering and outlier detection
tree_HTN <- hclust(dist(t(exprs(datAll_HTN))), method = "average")
plot(tree_HTN, main = "Hierarchical clustering of GSE24752 samples", xlab = "", sub = "")
```



```
normadj_HTN <- (0.5 + 0.5*bicor(exprs(datAll_HTN)))^2
netsummary_HTN <- fundamentalNetworkConcepts(normadj_HTN)
C_HTN <- netsummary_HTN$Connectivity
Z.C_HTN <- (C_HTN - mean(C_HTN)) / sqrt(var(C_HTN))

datLabel_HTN <- pData(datAll_HTN)$Dx
plot(1:length(Z.C_HTN),Z.C_HTN,main="Outlier plot of GSE24752 samples ",xlab = "Samples",ylab="Connecti
text(1:length(Z.C_HTN),Z.C_HTN,label=datLabel_HTN,pos=3,cex=0.8)
abline(h= -2, col="red", lwd = 2)
```



```r
# Identify and remove potential outlier from GSE24752 samples based on connectivity Z-score
# No samples exceeded the threshold (Z < -2), so none were removed
to_keep_HTN <- abs(Z.C_HTN) < 2
table(to_keep_HTN)
```

```
## to_keep_HTN
## FALSE  TRUE
##     1     5
```

```r
colnames(exprs(datAll_HTN))[!to_keep_HTN]
```

```
## [1] "GSM609525"
```

```r
datAll_HTN <- datAll_HTN[, to_keep_HTN]
```

```r
# Annotating Probes using Ensembl
ensembl <- useEnsembl(biomart = "ensembl", dataset = "hsapiens_gene_ensembl")
```

```r
# Annotating Probes for GSE24752 dataset
identifier_HTN <- "affy_hg_u133_plus_2"
getinfo_HTN <- c("affy_hg_u133_plus_2", "ensembl_gene_id", "entrezgene_id", "external_gene_name")
geneDat_HTN <- getBM(attributes = getinfo_HTN,
                     filters = identifier_HTN,
                      values = rownames(exprs(datAll_HTN)),
                      mart = ensembl)
idx_HTN <- match(rownames(exprs(datAll_HTN)), geneDat_HTN$affy_hg_u133_plus_2)
geneDat_HTN <- geneDat_HTN[idx_HTN, ]
table(is.na(geneDat_HTN$ensembl_gene_id))
```

```
##
## FALSE  TRUE
## 43709 10966
```

```
to_keep_HTN <- !is.na(geneDat_HTN$ensembl_gene_id)
geneDat_HTN <- geneDat_HTN[to_keep_HTN, ]
datAll_HTN <- datAll_HTN[to_keep_HTN, ]
```

```
# Collapse Rows for GSE24752 by Ensembl Gene ID
table(duplicated(geneDat_HTN$affy_hg_u133_plus_2))
```

```
##
## FALSE
## 43709
```

```
table(duplicated(geneDat_HTN$ensembl_gene_id))
```

```
##
## FALSE  TRUE
## 23648 20061
```

```
CR_HTN <- collapseRows(exprs(datAll_HTN),
                        rowGroup = geneDat_HTN$ensembl_gene_id,
                        rowID = geneDat_HTN$affy_hg_u133_plus_2)
```

```
## Warning: 5 or fewer samples, this method of probe collapse is unreliable...
## ...Running anyway, but we suggest trying another method (for example, *mean*).
```

```
CRdata_HTN <- CR_HTN$datETcollapsed
idx_HTN <- match(CR_HTN$group2row[,"selectedRowID"], geneDat_HTN$affy_hg_u133_plus_2)
geneDat_HTN <- geneDat_HTN[idx_HTN, ]
rownames(geneDat_HTN) <- geneDat_HTN$ensembl_gene_id
```

```
# Load a GSE28345 csv file
data_GSE28345 <- read.csv("/Users/lorandacalderonzamora/Downloads/Normalized_matriz_GSE28345", row.name
```

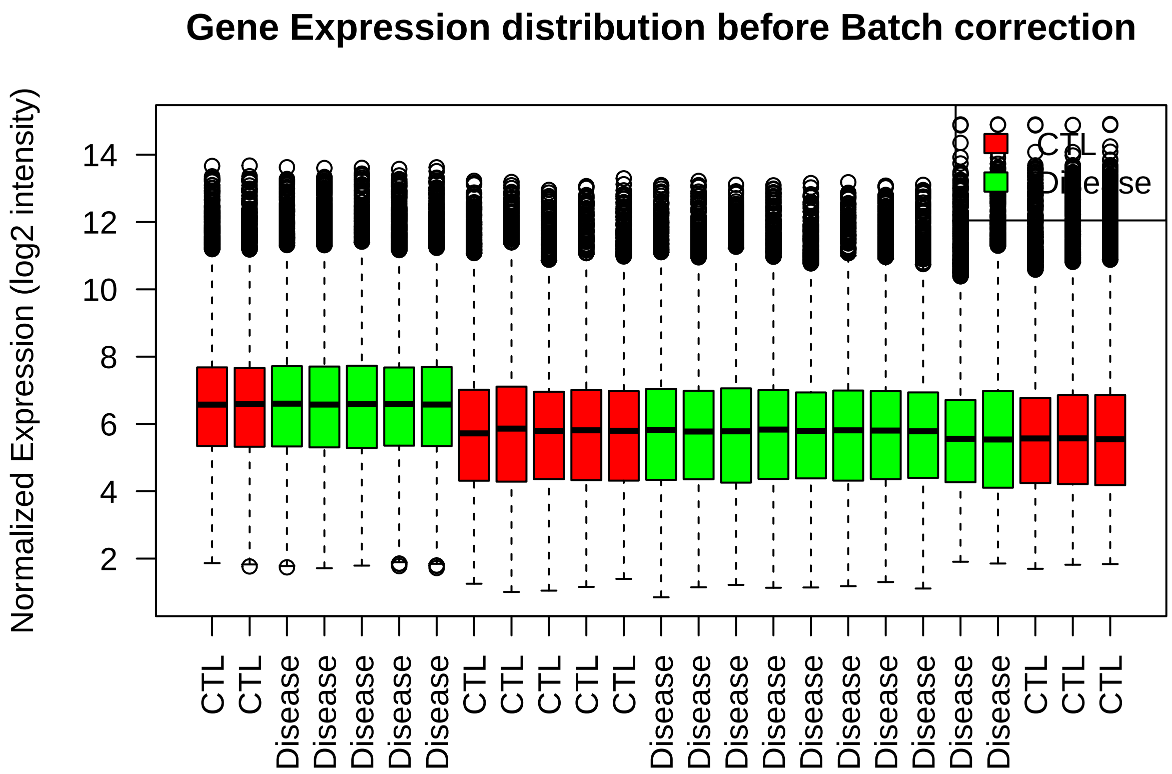
```
# Load a GSE28360 csv file
data_GSE28360 <- read.csv("/Users/lorandacalderonzamora/Downloads/datExpr_GSE28360.csv", row.names = 1)
```

```
# Merging GSE24752, GSE28345 and GSE28360 Expression profiles by common genes
common_genes <- intersect(intersect(rownames(data_GSE28345), rownames(data_GSE28360)),
  rownames(CRdata_HTN))
```

```
data_GSE28345_common <- data_GSE28345[common_genes, ]
data_GSE28360_common <- data_GSE28360[common_genes, ]
CRdata_HTN_common     <- CRdata_HTN[common_genes, ]

unificated_expr_matrix <- cbind(data_GSE28345_common,data_GSE28360_common,CRdata_HTN_common)
unificated_expr_matrix <- as.data.frame(unificated_expr_matrix)
```

```
# Relabeling sample identifiers with group labels for Differential Expression Analysis
sample_ids <- colnames(unificated_expr_matrix)

group_labels <- sample_ids
group_labels[group_labels %in% c("GSM700797", "GSM700798", "GSM609528", "GSM609529",
                                 "GSM609530", "GSM701161", "GSM701162", "GSM701163", "GSM701164", "GSM7

group_labels[group_labels %in% c("GSM700799", "GSM700800", "GSM700801", "GSM700802", "GSM700803",
                                 "GSM609526", "GSM609527", "GSM701166", "GSM701167", "GSM701168", "GSM7

colnames(unificated_expr_matrix) <- group_labels
```

```
# Boxplot of the merged Expression matrix GSE24752, GSE28345 and GSE28360 prior to Batch correction
boxplot(unificated_expr_matrix, main = "Gene Expression distribution before Batch correction", col = c(
legend("topright", legend = levels(factor(group_labels)), fill = c("red", "green"))
```



```r
# Quantile normalization of merged Expression matrix
library(preprocessCore)
expr_matrix_qn <- normalize.quantiles(as.matrix(unificated_expr_matrix))
rownames(expr_matrix_qn) <- rownames(unificated_expr_matrix)
colnames(expr_matrix_qn) <- colnames(unificated_expr_matrix)
```

```r
# Batch effect correction between GSE24752, GSE28345 and GSE28360 Datasets using ComBat
n_GSE24752 <- ncol(CRdata_HTN_common)
n_GSE28345 <- ncol(data_GSE28345_common)
n_GSE28360 <- ncol(data_GSE28360_common)
batch <- c(rep("GSE24752", n_GSE24752), rep("GSE28345", n_GSE28345), rep("GSE28360", n_GSE28360))

combat_expr <- ComBat(dat = expr_matrix_qn, batch = batch, par.prior = TRUE, prior.plots = FALSE)
```

```
## Found 1 genes with uniform expression within a single batch (all zeros); these will not be adjusted
```

```
## Found3batches
```

```
## Adjusting for0covariate(s) or covariate level(s)
```

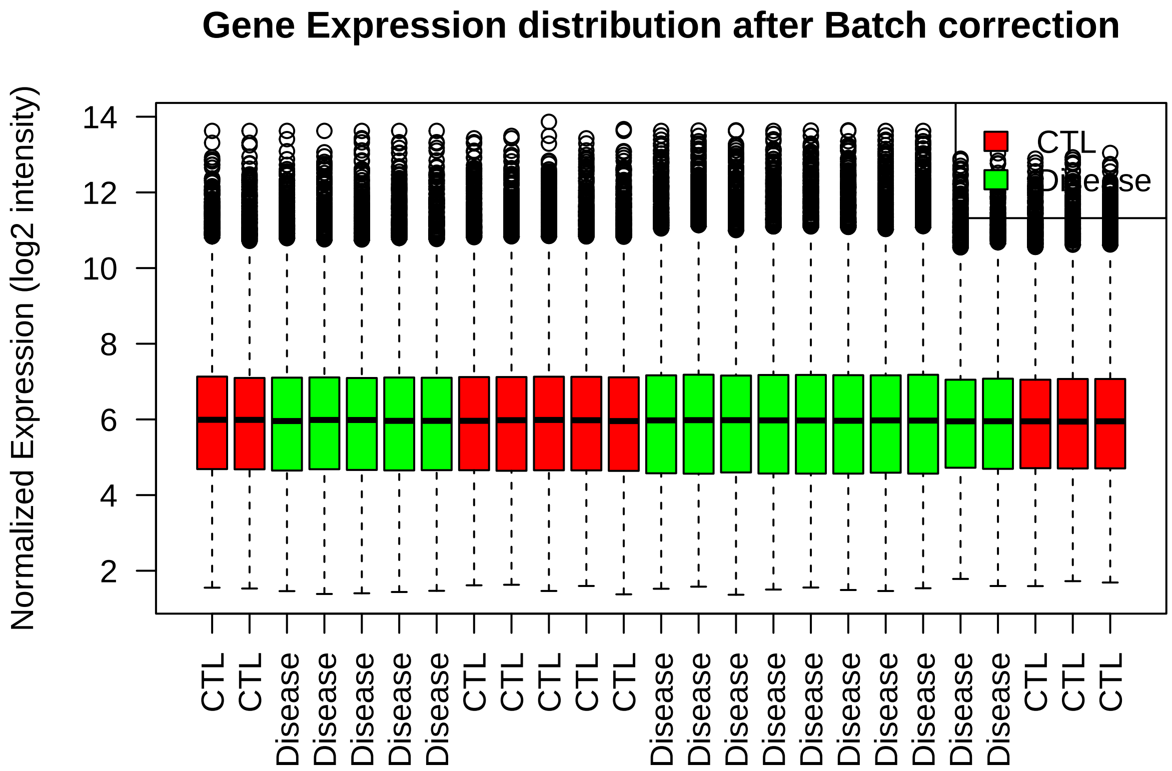
```
## Standardizing Data across genes
```

```
## Fitting L/S model and finding priors
```

```
## Finding parametric adjustments
```

```
## Adjusting the Data
```

```r
# Boxplot of the merged Expression matrix GSE24752, GSE28345 and GSE28360 after to Batch correction
boxplot(combat_expr, main = "Gene Expression distribution after Batch correction", col = c("red", "gree
legend("topright", legend = levels(factor(group_labels)), fill = c("red", "green"))
```



```r
# Define group factor for Differential Expression analysis
sample_ids <- colnames(combat_expr)
group <- as.factor(colnames(combat_expr))
```

```r
# Differential Expression Analysis between Control and Disease samples from integrated GSE24752 and GSE
design <- model.matrix(~ group)
fit_HTN <- lmFit(combat_expr, design)
fit_HTN <- eBayes(fit_HTN)
tt_HTN <- topTable(fit_HTN, coef = 2, number = Inf)
head(tt_HTN)
```

```
##                      logFC  AveExpr         t      P.Value adj.P.Val
## ENSG00000116717  0.6508212 7.757558  4.880065 4.872194e-05 0.2518421
## ENSG00000137959 -0.7986912 6.351395 -4.418920 1.625602e-04 0.2518421
## ENSG00000183793 -0.3803476 8.513381 -4.302510 2.202160e-04 0.2518421
## ENSG00000174652 -0.6184843 6.754409 -4.237603 2.607627e-04 0.2518421
## ENSG00000198625 -0.4944385 7.596977 -4.175117 3.067686e-04 0.2518421
## ENSG00000112941 -0.5109129 6.869815 -4.170446 3.105147e-04 0.2518421
##                           B
## ENSG00000116717  1.16574862
## ENSG00000137959  0.32161833
## ENSG00000183793  0.10640804
## ENSG00000174652 -0.01380893
## ENSG00000198625 -0.12964741
## ENSG00000112941 -0.13831069
```

```r
# Annotating Differential Expression using Ensembl gene IDs
gene_ids <- rownames(tt_HTN)
annot_attributes <- c("ensembl_gene_id", "external_gene_name", "entrezgene_id")
geneDat <- getBM(attributes = annot_attributes,
                 filters = "ensembl_gene_id",
                 values = gene_ids,
                 mart = ensembl)

tt_HTN$ensembl_gene_id <- rownames(tt_HTN)
tt_annotated <- merge(tt_HTN, geneDat, by = "ensembl_gene_id")

tt_annotated <- tt_annotated[, c("ensembl_gene_id", "external_gene_name", "entrezgene_id",
                                 "logFC", "AveExpr", "t", "P.Value", "adj.P.Val", "B")]

head(tt_annotated)
```

```
##   ensembl_gene_id external_gene_name entrezgene_id       logFC  AveExpr
## 1 ENSG00000000003             TSPAN6          7105  0.51178060 7.508559
## 2 ENSG00000000005               TNMD         64102  0.02368208 2.925032
## 3 ENSG00000000419               DPM1          8813 -0.09931190 8.284134
## 4 ENSG00000000457              SCYL3         57147 -0.11700606 6.633253
## 5 ENSG00000000460              FIRRM         55732  0.11538083 3.876373
## 6 ENSG00000000938                FGR          2268 -0.52440971 6.630339
##            t   P.Value adj.P.Val         B
## 1  0.9415992 0.3552573 0.5954354 -5.065927
## 2  0.3522479 0.7275548 0.8548354 -5.380575
## 3 -0.8857048 0.3840755 0.6193039 -5.107438
## 4 -1.0683656 0.2953927 0.5434598 -4.963202
## 5  0.5814721 0.5660413 0.7521183 -5.291141
## 6 -1.2417106 0.2256743 0.4796944 -4.804113
```