

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

Loranda\_Calderon

2025-06-01

```
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)
```

```
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(ensembladb)
```

```
## 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)
```

```

# 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)

# 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)

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

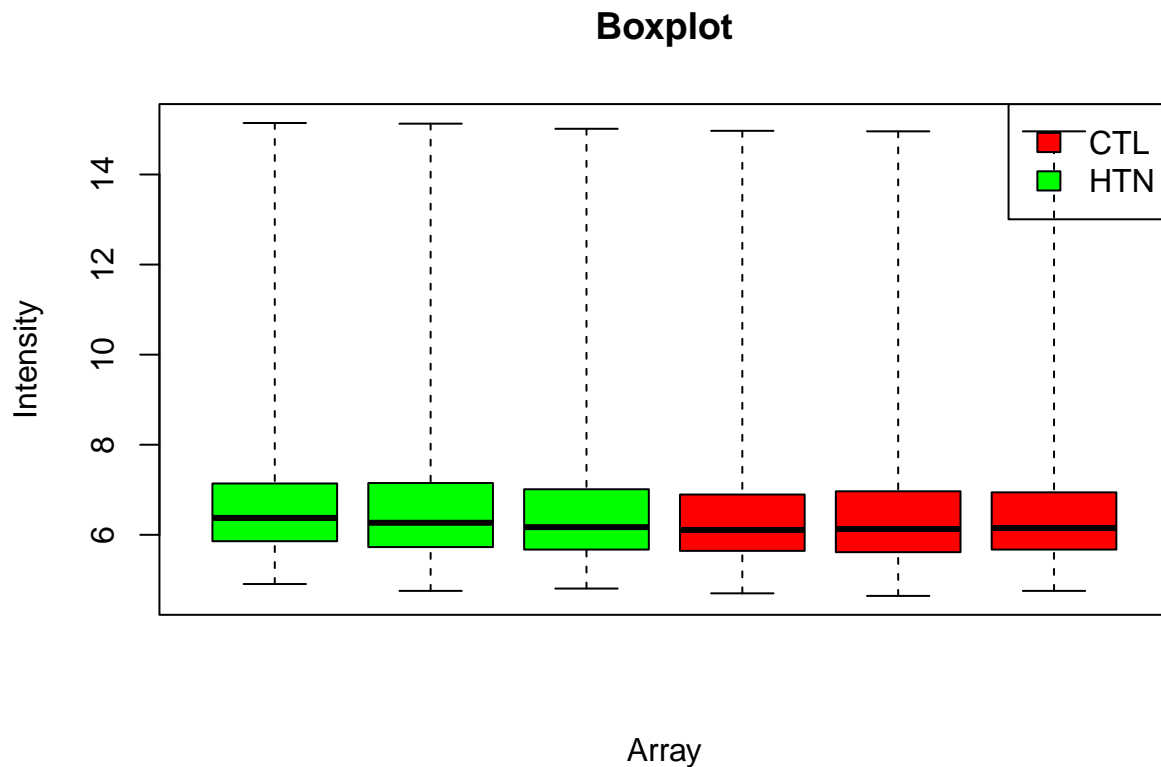
```

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

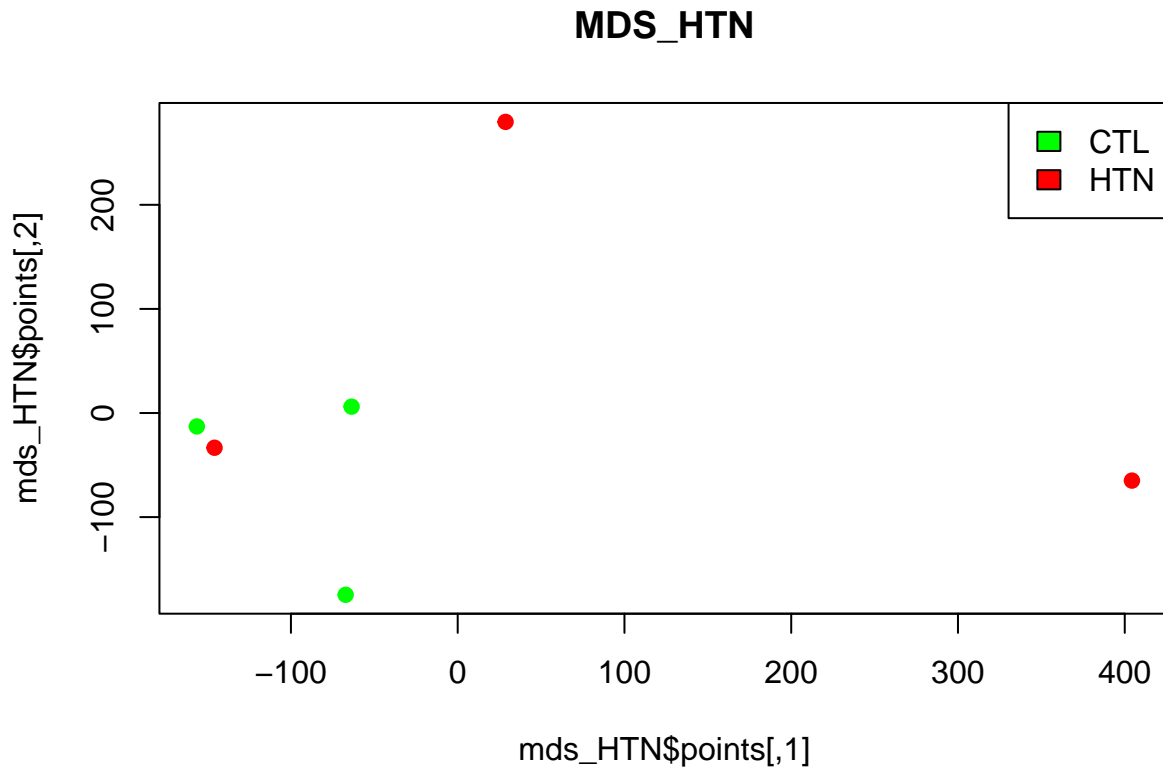
```

# 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(d
```



```
# 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'
```

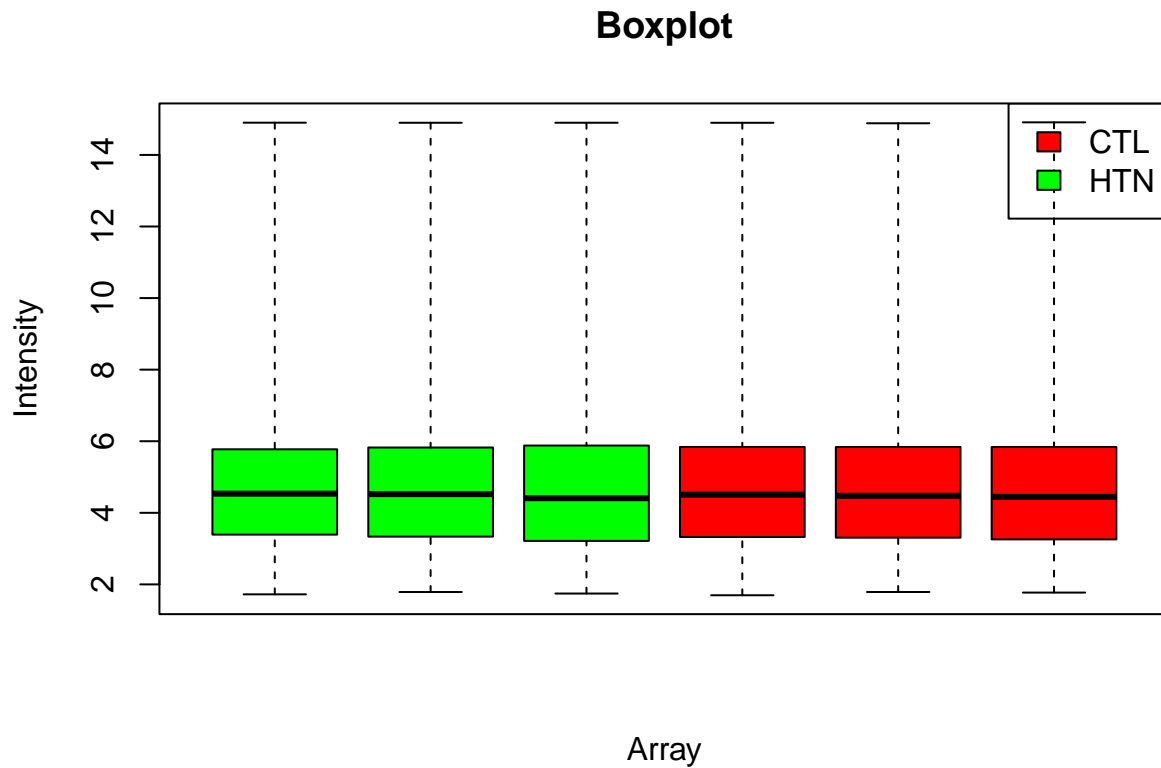
```
## Warning: replacing previous import 'AnnotationDbi::head' by 'utils::head' when
## loading 'hgu133plus2cdf'
```

```
##
```

```
## 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(
```



```
# 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)
```

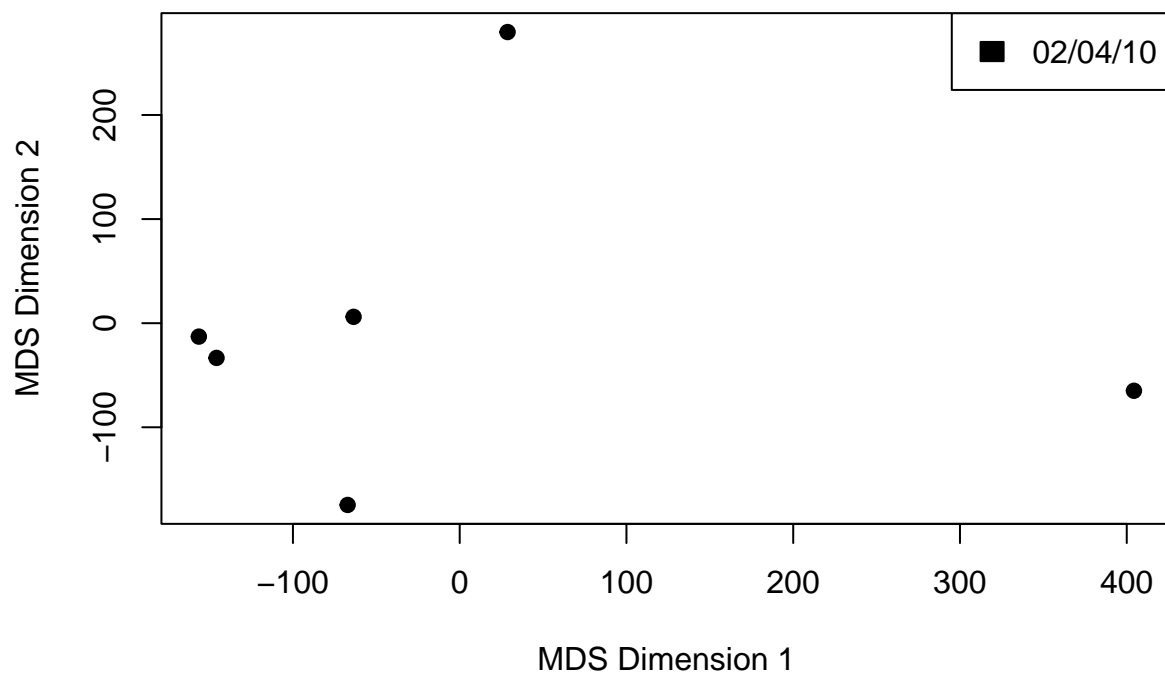
```
# Extract ScanDate from GSE24752 for batch effect correction
batch_HTN <- protocolData(data.affy_HTN)$ScanDate
batch_HTN <- substr(batch_HTN,1,8)
batch_HTN <- as.factor(batch_HTN)
table(batch_HTN)
```

```
## 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 Batch",
legend("topright",legend = levels(datMeta_HTN$Batch),fill = as.numeric(as.factor(levels(datMeta_HTN$Batch))))
```

## MDS Plot of GSE24752 Colored by Batch

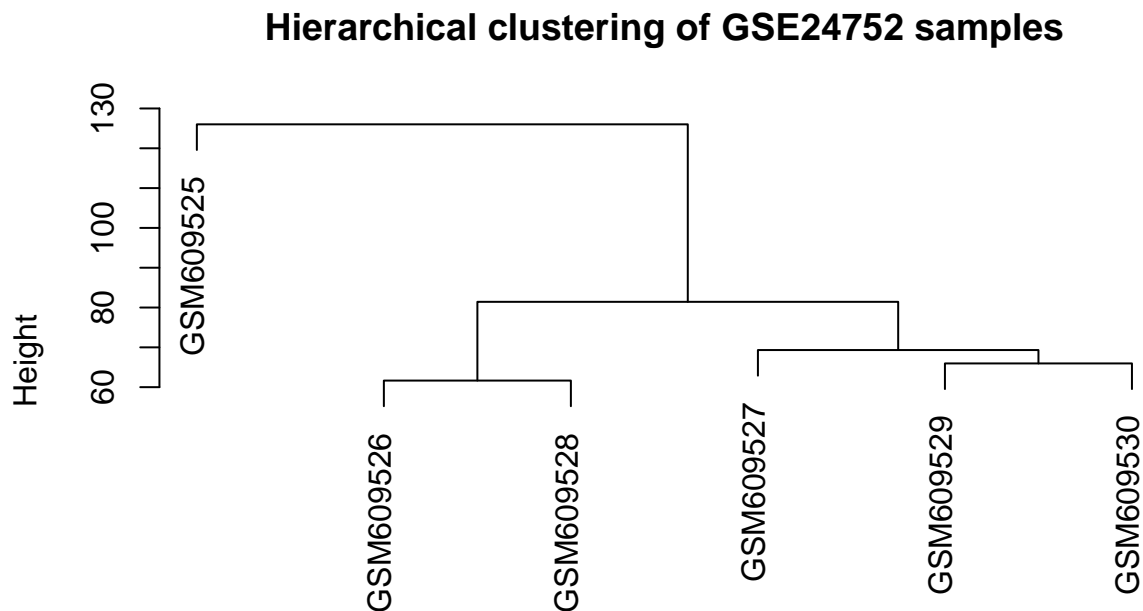


```

# 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 = "")

```



```

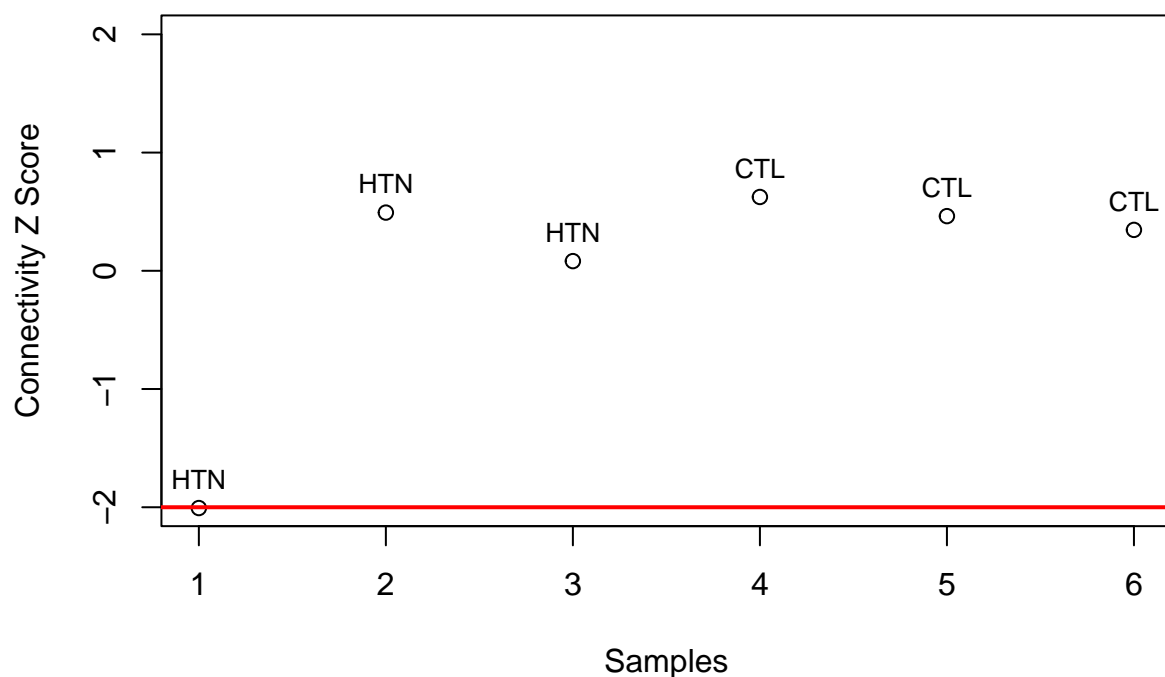
normadj_HTN <- (0.5 + 0.5*bicor(exprs(datAll_HTN)))^2
netsummary_HTN <- fundamentalNetworkConcepts(normadj_HTN)
C_HTN <- netsummary_HTN$Connectivity
Z.C_HTN <- (C_HTN - mean(C_HTN)) / sqrt(var(C_HTN))

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

```



## Outlier plot of GSE24752 samples



```
# 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
```

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

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

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

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

```
# 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.names = 1)

```

```

# 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))

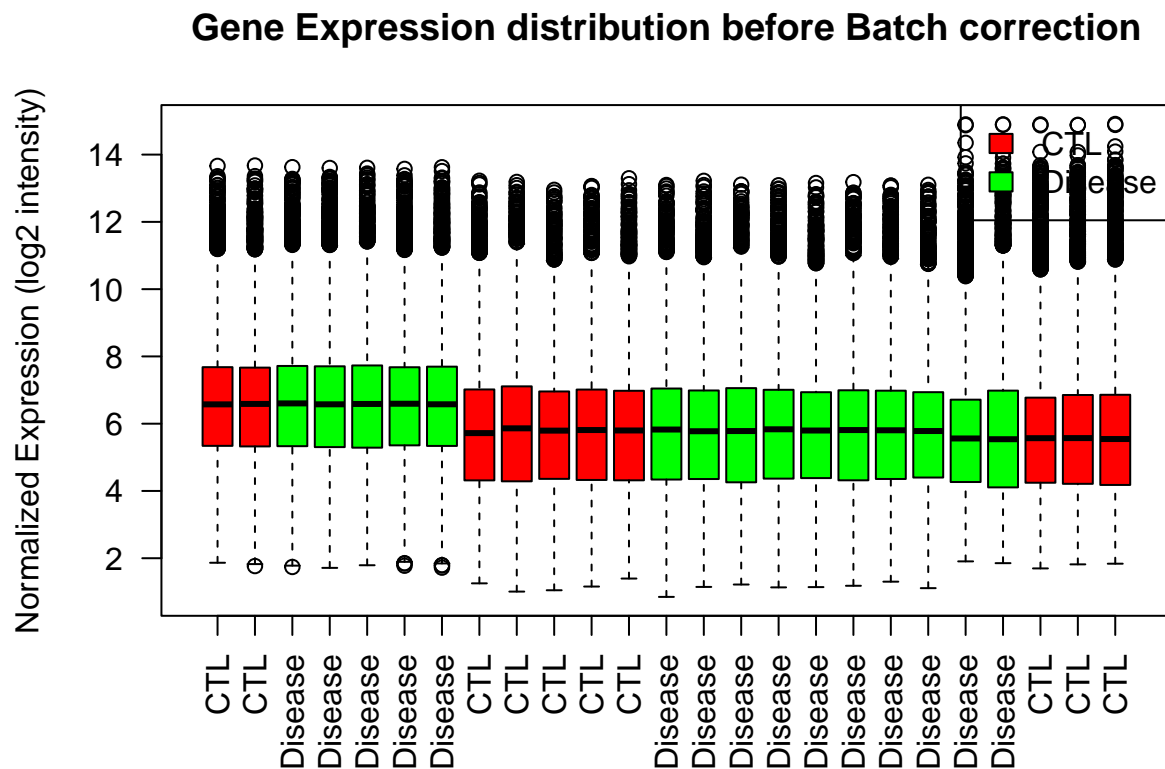
```

```
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", "GSM701165",
                                "GSM701166", "GSM701167", "GSM701168", "GSM701169", "GSM701170", "GSM701171", "GSM701172", "GSM701173", "GSM701174", "GSM701175", "GSM701176", "GSM701177", "GSM701178", "GSM701179", "GSM701180", "GSM701181", "GSM701182", "GSM701183", "GSM701184", "GSM701185", "GSM701186", "GSM701187", "GSM701188", "GSM701189", "GSM701190", "GSM701191", "GSM701192", "GSM701193", "GSM701194", "GSM701195", "GSM701196", "GSM701197", "GSM701198", "GSM701199", "GSM701200", "GSM701201", "GSM701202", "GSM701203", "GSM701204", "GSM701205", "GSM701206", "GSM701207", "GSM701208", "GSM701209", "GSM701210", "GSM701211", "GSM701212", "GSM701213", "GSM701214", "GSM701215", "GSM701216", "GSM701217", "GSM701218", "GSM701219", "GSM701220", "GSM701221", "GSM701222", "GSM701223", "GSM701224", "GSM701225", "GSM701226", "GSM701227", "GSM701228", "GSM701229", "GSM701230", "GSM701231", "GSM701232", "GSM701233", "GSM701234", "GSM701235", "GSM701236", "GSM701237", "GSM701238", "GSM701239", "GSM701240", "GSM701241", "GSM701242", "GSM701243", "GSM701244", "GSM701245", "GSM701246", "GSM701247", "GSM701248", "GSM701249", "GSM701250", "GSM701251", "GSM701252", "GSM701253", "GSM701254", "GSM701255", "GSM701256", "GSM701257", "GSM701258", "GSM701259", "GSM701260", "GSM701261", "GSM701262", "GSM701263", "GSM701264", "GSM701265", "GSM701266", "GSM701267", "GSM701268", "GSM701269", "GSM701270", "GSM701271", "GSM701272", "GSM701273", "GSM701274", "GSM701275", "GSM701276", "GSM701277", "GSM701278", "GSM701279", "GSM701280", "GSM701281", "GSM701282", "GSM701283", "GSM701284", "GSM701285", "GSM701286", "GSM701287", "GSM701288", "GSM701289", "GSM701290", "GSM701291", "GSM701292", "GSM701293", "GSM701294", "GSM701295", "GSM701296", "GSM701297", "GSM701298", "GSM701299", "GSM701300", "GSM701301", "GSM701302", "GSM701303", "GSM701304", "GSM701305", "GSM701306", "GSM701307", "GSM701308", "GSM701309", "GSM701310", "GSM701311", "GSM701312", "GSM701313", "GSM701314", "GSM701315", "GSM701316", "GSM701317", "GSM701318", "GSM701319", "GSM701320", "GSM701321", "GSM701322", "GSM701323", "GSM701324", "GSM701325", "GSM701326", "GSM701327", "GSM701328", "GSM701329", "GSM701330", "GSM701331", "GSM701332", "GSM701333", "GSM701334", "GSM701335", "GSM701336", "GSM701337", "GSM701338", "GSM701339", "GSM701340", "GSM701341", "GSM701342", "GSM701343", "GSM701344", "GSM701345", "GSM701346", "GSM701347", "GSM701348", "GSM701349", "GSM701350", "GSM701351", "GSM701352", "GSM701353", "GSM701354", "GSM701355", "GSM701356", "GSM701357", "GSM701358", "GSM701359", "GSM701360", "GSM701361", "GSM701362", "GSM701363", "GSM701364", "GSM701365", "GSM701366", "GSM701367", "GSM701368", "GSM701369", "GSM701370", "GSM701371", "GSM701372", "GSM701373", "GSM701374", "GSM701375", "GSM701376", "GSM701377", "GSM701378", "GSM701379", "GSM701380", "GSM701381", "GSM701382", "GSM701383", "GSM701384", "GSM701385", "GSM701386", "GSM701387", "GSM701388", "GSM701389", "GSM701390", "GSM701391", "GSM701392", "GSM701393", "GSM701394", "GSM701395", "GSM701396", "GSM701397", "GSM701398", "GSM701399", "GSM701400", "GSM701401", "GSM701402", "GSM701403", "GSM701404", "GSM701405", "GSM701406", "GSM701407", "GSM701408", "GSM701409", "GSM701410", "GSM701411", "GSM701412", "GSM701413", "GSM701414", "GSM701415", "GSM701416", "GSM701417", "GSM701418", "GSM701419", "GSM701420", "GSM701421", "GSM701422", "GSM701423", "GSM701424", "GSM701425", "GSM701426", "GSM701427", "GSM701428", "GSM701429", "GSM701430", "GSM701431", "GSM701432", "GSM701433", "GSM701434", "GSM701435", "GSM701436", "GSM701437", "GSM701438", "GSM701439", "GSM701440", "GSM701441", "GSM701442", "GSM701443", "GSM701444", "GSM701445", "GSM701446", "GSM701447", "GSM701448", "GSM701449", "GSM701450", "GSM701451", "GSM701452", "GSM701453", "GSM701454", "GSM701455", "GSM701456", "GSM701457", "GSM701458", "GSM701459", "GSM701460", "GSM701461", "GSM701462", "GSM701463", "GSM701464", "GSM701465", "GSM701466", "GSM701467", "GSM701468", "GSM701469", "GSM701470", "GSM701471", "GSM701472", "GSM701473", "GSM701474", "GSM701475", "GSM701476", "GSM701477", "GSM701478", "GSM701479", "GSM701480", "GSM701481", "GSM701482", "GSM701483", "GSM701484", "GSM701485", "GSM701486", "GSM701487", "GSM701488", "GSM701489", "GSM701490", "GSM701491", "GSM701492", "GSM701493", "GSM701494", "GSM701495", "GSM701496", "GSM701497", "GSM701498", "GSM701499", "GSM701500", "GSM701501", "GSM701502", "GSM701503", "GSM701504", "GSM701505", "GSM701506", "GSM701507", "GSM701508", "GSM701509", "GSM701510", "GSM701511", "GSM701512", "GSM701513", "GSM701514", "GSM701515", "GSM701516", "GSM701517", "GSM701518", "GSM701519", "GSM701520", "GSM701521", "GSM701522", "GSM701523", "GSM701524", "GSM701525", "GSM701526", "GSM701527", "GSM701528", "GSM701529", "GSM701530", "GSM701531", "GSM701532", "GSM701533", "GSM701534", "GSM701535", "GSM701536", "GSM701537", "GSM701538", "GSM701539", "GSM701540", "GSM701541", "GSM701542", "GSM701543", "GSM701544", "GSM701545", "GSM701546", "GSM701547", "
```



```

# 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)

# 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

## Found 3 batches

## Adjusting for 0 covariate(s) or covariate level(s)

## Standardizing Data across genes

## Fitting L/S model and finding priors

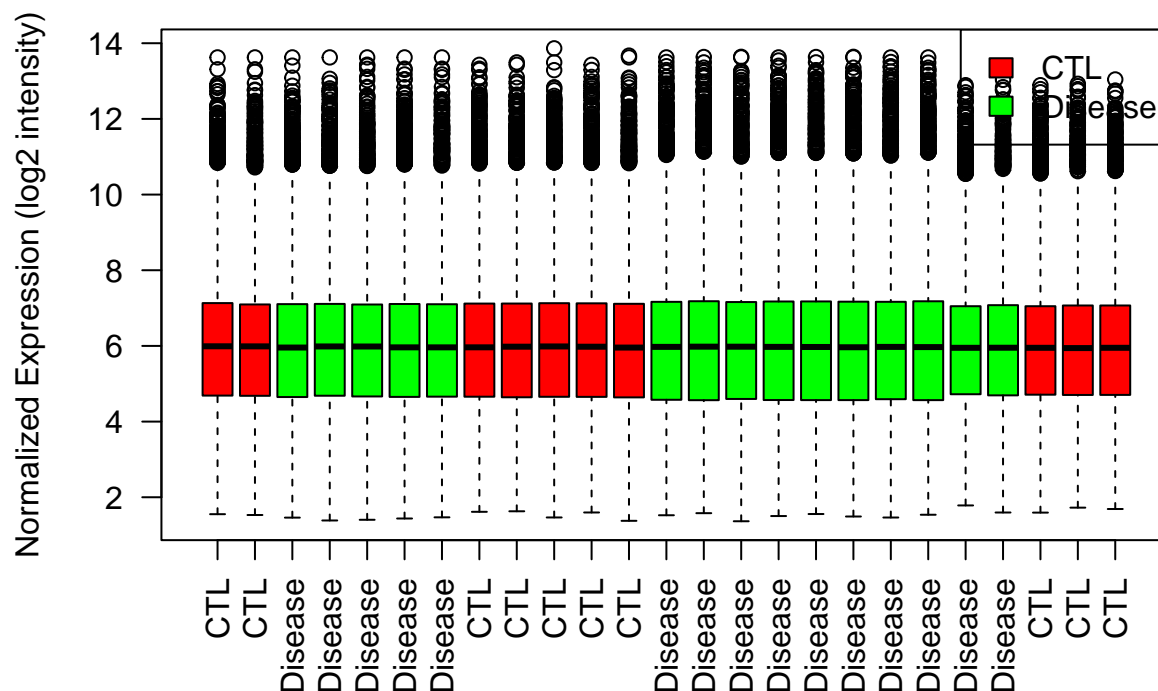
## Finding parametric adjustments

## Adjusting the Data

# 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", "green"),
legend("topright", legend = levels(factor(group_labels)), fill = c("red", "green"))

```

## Gene Expression distribution after Batch correction



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

```
# 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
```

```

# 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 ENSG000000000419          DPM1            8813 -0.09931190 8.284134
## 4 ENSG000000000457          SCYL3          57147 -0.11700606 6.633253
## 5 ENSG000000000460          FIRRM          55732  0.11538083 3.876373
## 6 ENSG000000000938           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

```