Transcriptomic preprocessing and Differential Expression analysis of GSE28345 (HTN Dataset)

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(oligo)
```

```
## Loading required package: oligoClasses

## Welcome to oligoClasses version 1.64.0
## Loading required package: Biostrings

## Loading required package: S4Vectors

## Loading required package: stats4

##
## 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

## Loading required package: XVector

## Loading required package: GenomeInfoDb

##
## Attaching package: 'Biostrings'

## The following object is masked from 'package:base':
##
##     strsplit

## ================================================================================

## Welcome to oligo version 1.66.0

## ================================================================================
```

```r
library(limma)
```

```
##
## Attaching package: 'limma'

## The following object is masked from 'package:oligo':
##
##     backgroundCorrect

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

```
library(sva)
```

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

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

```
##
## Attaching package: ’nlme’
```

```
## The following object is masked from ’package:Biostrings’:
##
##     collapse
```

```
## The following object is masked from ’package:IRanges’:
##
##     collapse
```

```
## 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:IRanges’:
##
##     cor
```

```
## The following object is masked from ’package:S4Vectors’:
##
##     cor
```

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

```r
library(ensembldb)
```

```
## Loading required package: GenomicRanges

## Loading required package: GenomicFeatures

## Loading required package: AnnotationDbi

## Loading required package: AnnotationFilter

##
## Attaching package: ’ensembldb’

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

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

```r
# Download GSE28345 data
gse_HTN <- getGEO("GSE28345", GSEMatrix =TRUE,getGPL=FALSE)
```

```
## Found 1 file(s)

## GSE28345_series_matrix.txt.gz
```

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

```r
# Read GSE28345 data
setwd("/Users/lorandacalderonzamora/GSE28345/")
celfiles <- list.files(pattern = ".CEL.gz$", full.names = TRUE)
rawData_HTN <- read.celfiles(celfiles)
```

```
## Loading required package: pd.hugene.1.0.st.v1

## Loading required package: RSQLite

## Loading required package: DBI

## Platform design info loaded.

## Reading in : ./GSM700796.CEL.gz
## Reading in : ./GSM700797.CEL.gz
## Reading in : ./GSM700798.CEL.gz
## Reading in : ./GSM700799.CEL.gz
## Reading in : ./GSM700800.CEL.gz
## Reading in : ./GSM700801.CEL.gz
## Reading in : ./GSM700802.CEL.gz
## Reading in : ./GSM700803.CEL.gz
```

```r
datExpr_HTN <- exprs(rawData_HTN)
```

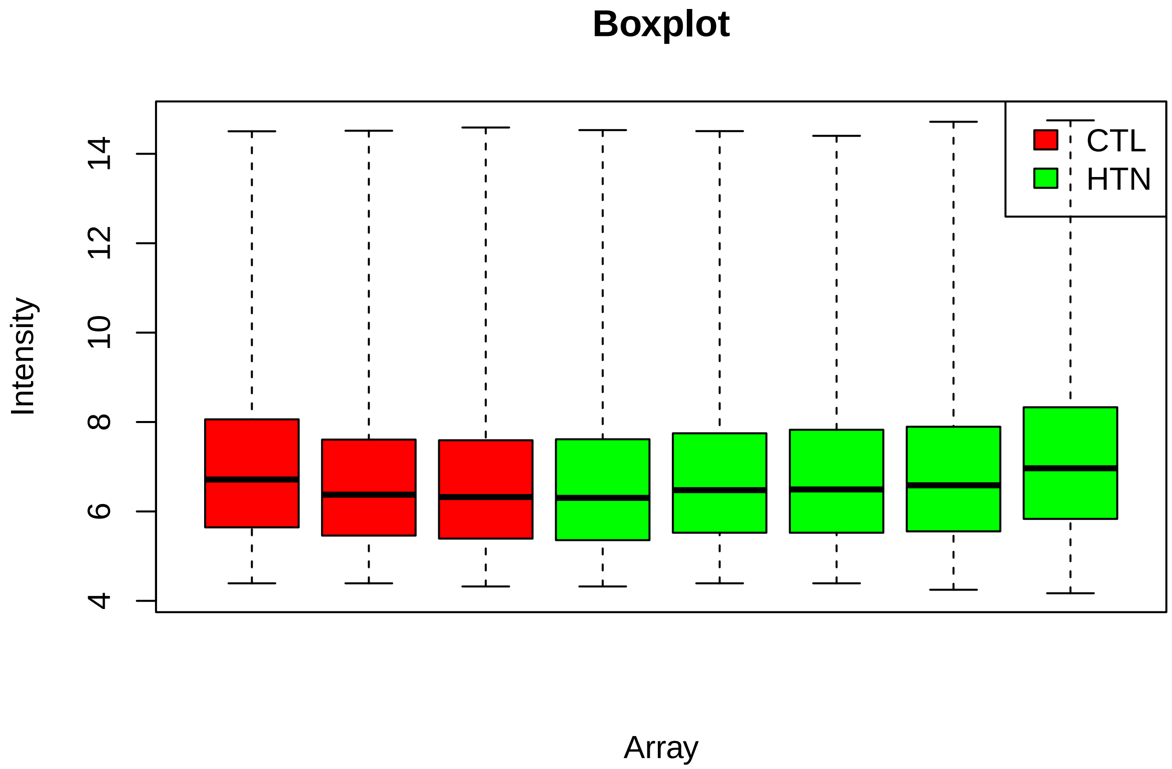
```r
# Align datMeta_HTN and datExpr_HTN by sample identifiers
GSM_HTN <- rownames(pData(rawData_HTN))
GSM_HTN <- substr(GSM_HTN, 1, 9)
idx_HTN <- match(GSM_HTN, datMeta_HTN$geo_accession)
datMeta_HTN <- datMeta_HTN[idx_HTN, ]
colnames(datExpr_HTN)=rownames(datMeta_HTN)
```

```r
# Cleaning and formatting of GSE28345 metadata
datMeta_HTN <- datMeta_HTN[, -c(3:7, 14:36)]
colnames(datMeta_HTN)[3] <- "Dx"
datMeta_HTN$Dx[rownames(datMeta_HTN) %in% c("GSM700796", "GSM700797", "GSM700798")] <- "CTL"
datMeta_HTN$Dx[rownames(datMeta_HTN) %in% c("GSM700799", "GSM700800", "GSM700801", "GSM700802", "GSM7008
datMeta_HTN$Dx <- as.factor(datMeta_HTN$Dx)
```

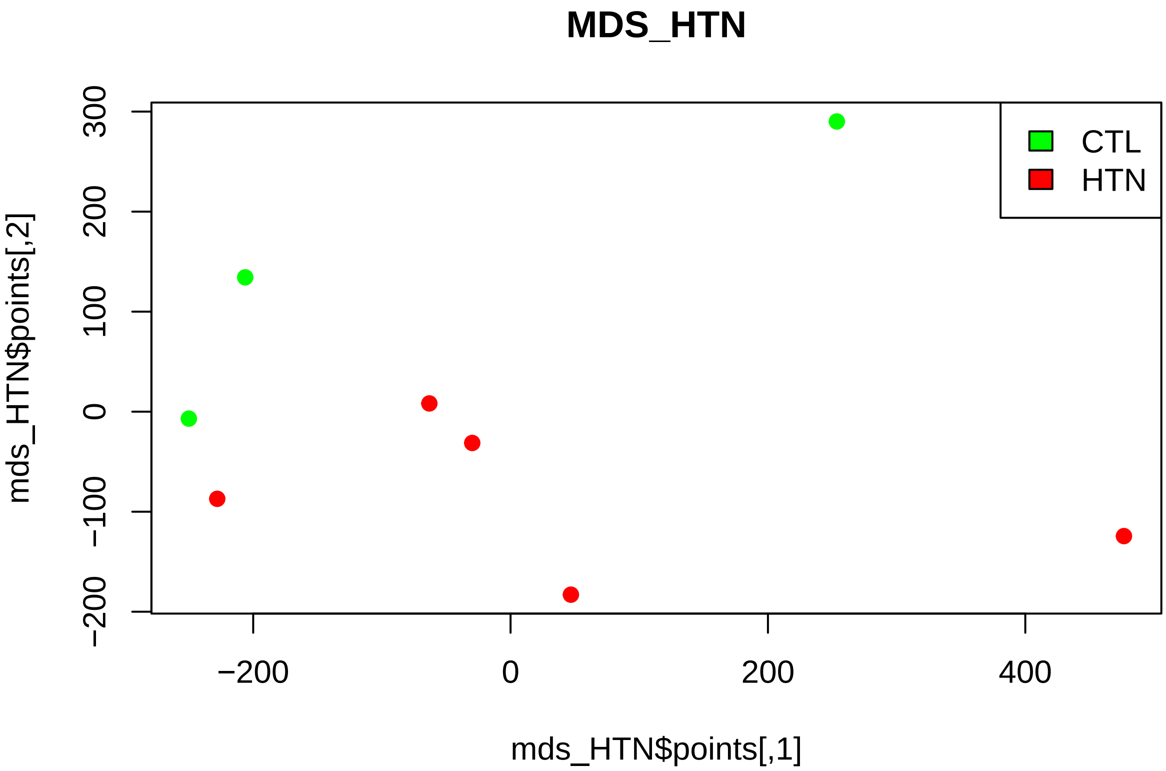
```r
# Preprocessing and quality assessment of GSE28345 raw expression data
datExpr_HTN <- log2(datExpr_HTN)
dim(datExpr_HTN)
```

```
## [1] 1102500       8
```

```r
# Exploratory visualization of GSE28345 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
```

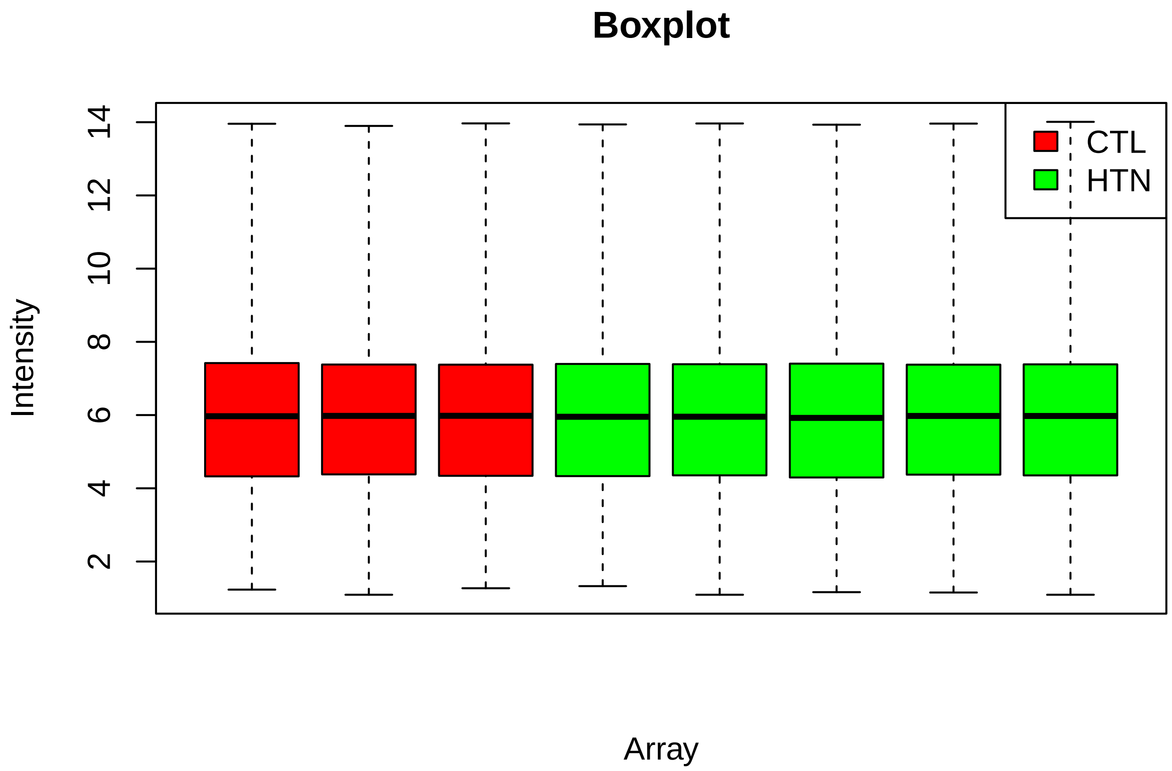


```
# Normalization using RMA
datExpr_HTN <- rma(rawData_HTN)
```

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

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

```
# Exploratory visualization of GSE28345 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
```



```r
# 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_GSE28345_Report",
                    force = TRUE,
                    do.logtransform = FALSE)
```

```
## The report will be written into directory '/Users/lorandacalderonzamora/Downloads/QC_GSE28345_Report
```

```
## 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
## 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
```

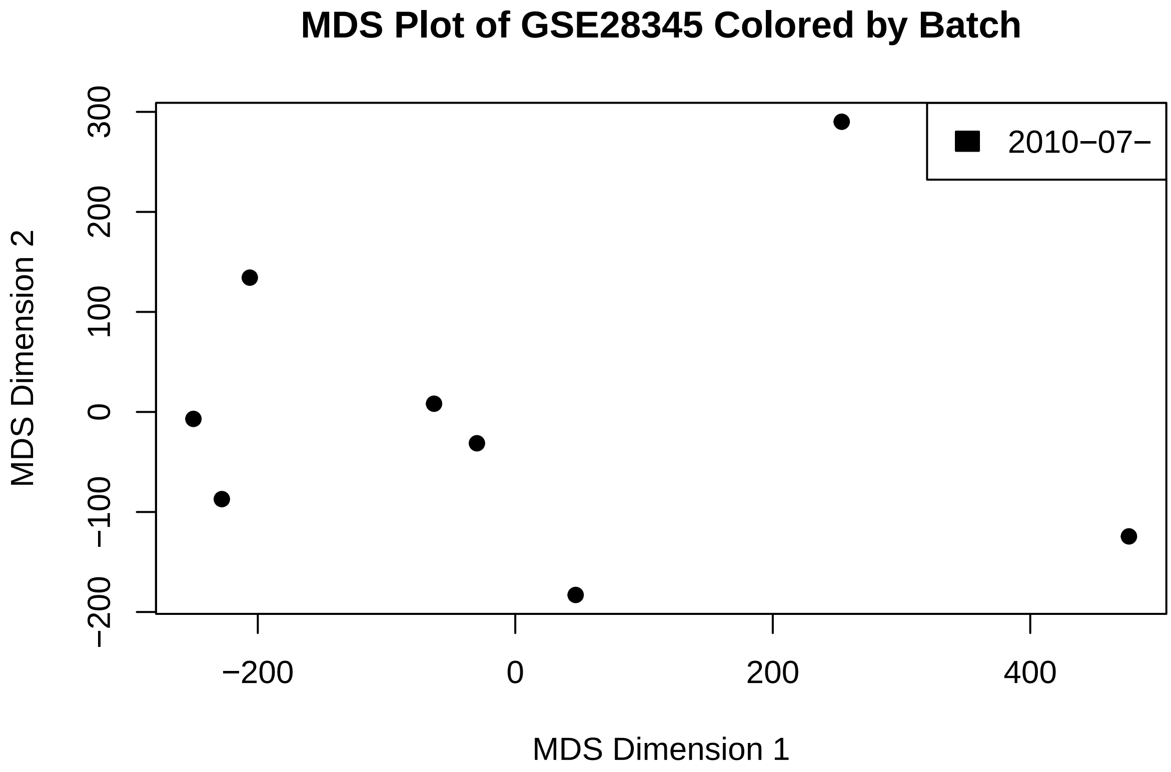
```
## (loaded the KernSmooth namespace)
```

```
# Extract ScanDate from GSE28345 for batch effect correction
batch_HTN <- protocolData(rawData_HTN)$dates
batch_HTN <- substr(batch_HTN,1,8)
batch_HTN <- as.factor(batch_HTN)
table(batch_HTN)
```

```
## batch_HTN
## 2010-07-
##        8
```

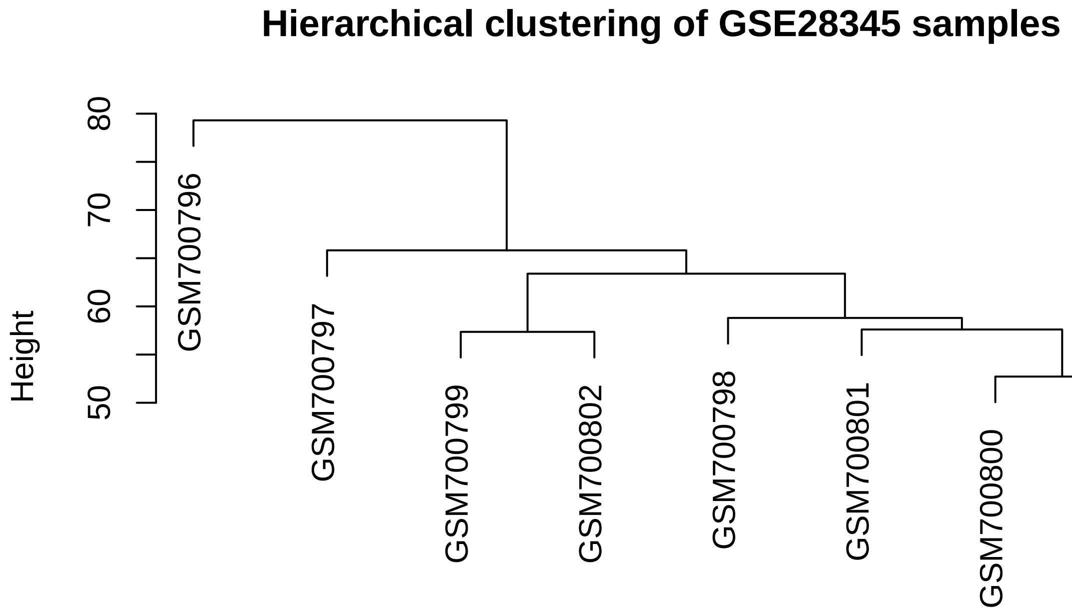
```
datMeta_HTN$Batch <- batch_HTN
```

```
# Visualization of ScanDate metadata from GSE28345 to identify potential batch effects
plot(mds_HTN$points,col = as.numeric(datMeta_HTN$Batch),pch=19,main="MDS Plot of GSE28345 Colored by Ba
legend("topright",legend = levels(datMeta_HTN$Batch),fill = as.numeric(as.factor(levels(datMeta_HTN$Bat
```



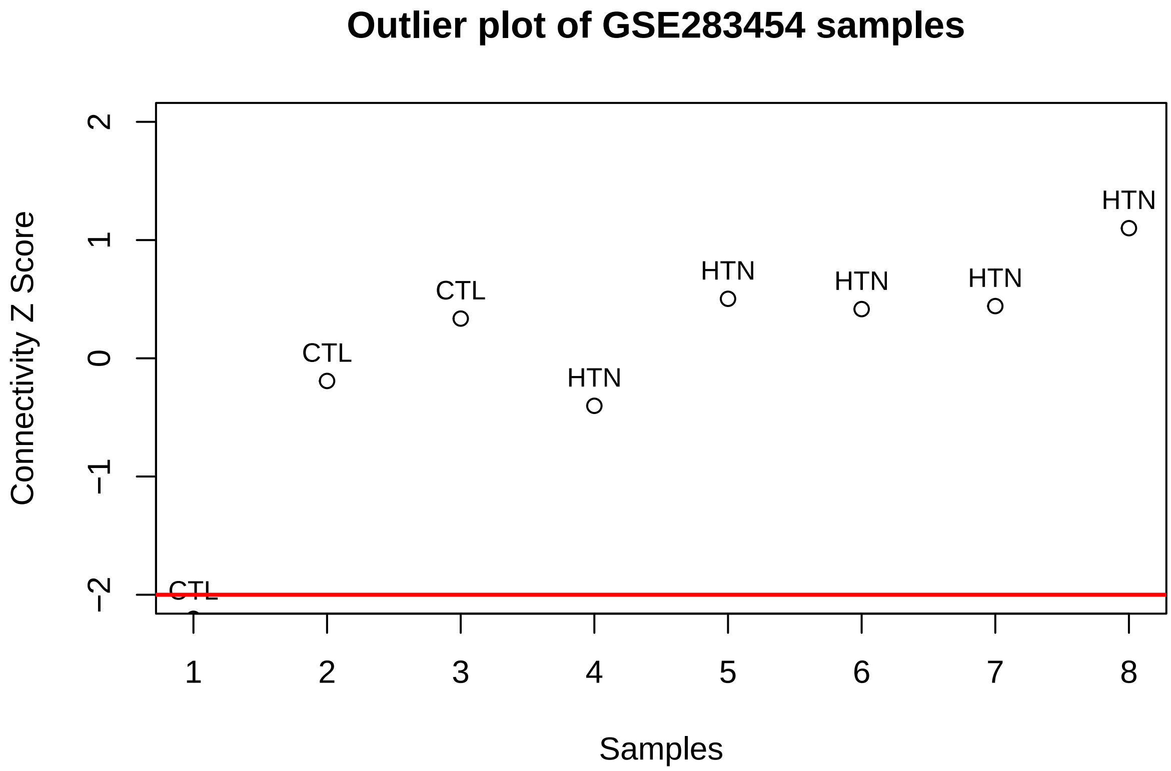
```r
# Create ExpressionSet object after Batch effect assessment
datMeta_HTN$Batch <- batch_HTN
datMeta_proc_HTN <- new("AnnotatedDataFrame", data = datMeta_HTN)
colnames(datExpr_HTN) <- rownames(datMeta_HTN)
datAll_HTN <- new("ExpressionSet", exprs = datExpr_HTN, phenoData = datMeta_proc_HTN)
# No singular batch was detected in the GSE28345 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.
```

```r
# Sample Clustering and outlier detection
tree_HTN <- hclust(dist(t(exprs(datAll_HTN))), method = "average")
plot(tree_HTN, main = "Hierarchical clustering of GSE28345 samples", xlab = "", sub = "")
```



```r
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 GSE283454 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 GSE28345 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     7
```

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

```
## [1] "GSM700796"
```

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

```r
# Annotating Probes for GSE28345 dataset
ensembl <- useEnsembl(biomart = "ensembl", dataset = "hsapiens_gene_ensembl")
identifier_HTN <- "affy_hugene_1_0_st_v1"
getinfo_HTN <- c("affy_hugene_1_0_st_v1", "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_hugene_1_0_st_v1)
```

```r
geneDat_HTN <- geneDat_HTN[idx_HTN, ]
table(is.na(geneDat_HTN$ensembl_gene_id))
```

```
##
## FALSE  TRUE
## 29120  4177
```

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

```r
# Collapse Rows for GSE28345 by Ensembl Gene ID
table(duplicated(geneDat_HTN$affy_hugene_1_0_st_v1))
```

```
##
## FALSE
## 29120
```

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

```
##
## FALSE  TRUE
## 24657  4463
```

```r
CR_HTN <- collapseRows(exprs(datAll_HTN),
                        rowGroup = geneDat_HTN$ensembl_gene_id,
                        rowID = geneDat_HTN$affy_hugene_1_0_st_v1)
CRdata_HTN <- CR_HTN$datETcollapsed
idx_HTN <- match(CR_HTN$group2row[,"selectedRowID"], geneDat_HTN$affy_hugene_1_0_st_v1)
geneDat_HTN <- geneDat_HTN[idx_HTN, ]
rownames(geneDat_HTN) <- geneDat_HTN$ensembl_gene_id
```

```r
# Differential Expression Analysis from GSE28345
mod <- model.matrix(~pData(datAll_HTN)$Dx)
fit <- lmFit(CR_HTN$datETcollapsed,mod)
fit <- eBayes(fit)
tt <- topTable(fit,coef = 2,n = Inf,genelist = geneDat_HTN)
head(tt)
```

```
##                 affy_hugene_1_0_st_v1 ensembl_gene_id entrezgene_id
## ENSG00000207475               7920873 ENSG00000207475        677823
## ENSG00000206836               8047215 ENSG00000206836            NA
## ENSG00000200935               7930775 ENSG00000200935            NA
## ENSG00000233901               8158539 ENSG00000233901     100506119
## ENSG00000112299               8129618 ENSG00000112299          8876
## ENSG00000296902               7919146 ENSG00000296902            NA
##                 external_gene_name      logFC  AveExpr         t      P.Value
## ENSG00000207475           SNORA80E -1.1813774 9.358679 -7.092428 5.637114e-05
## ENSG00000206836         RNU6-1029P -1.1623262 3.621035 -6.337113 1.333561e-04
## ENSG00000200935         RNU6-1090P -0.6306511 1.981581 -6.109341 1.752811e-04
## ENSG00000233901          LINC01503 -0.5032218 5.425712 -5.969319 2.080465e-04
## ENSG00000112299               VNN1  1.1868030 7.915618  5.830749 2.471261e-04
## ENSG00000296902                    -0.5632943 4.806523 -5.758829 2.704947e-04
##                 adj.P.Val          B
## ENSG00000207475 0.6554611 -0.1852593
## ENSG00000206836 0.6554611 -0.5036279
## ENSG00000200935 0.6554611 -0.6127839
## ENSG00000233901 0.6554611 -0.6832315
## ENSG00000112299 0.6554611 -0.7555690
## ENSG00000296902 0.6554611 -0.7941720
```