Integrative Expression matrix from GSE20966 and GSE25754 for transcriptomic analysis of Type 2 Diabetes

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(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 GSE20966 data ()
gse_T2DM <- getGEO("GSE20966", GSEMatrix =TRUE,getGPL=FALSE)
```

```
## Found 1 file(s)

## GSE20966_series_matrix.txt.gz
```

```
datMeta_T2DM <- pData(gse_T2DM[[1]])
rownames(datMeta_T2DM) <- datMeta_T2DM$geo_accession
```

```
# Read GSE20966 data
setwd("/Users/lorandacalderonzamora/GSE20966/")
data.affy_T2DM <- ReadAffy(celfile.path = "./")
datExpr_T2DM <- exprs(data.affy_T2DM)
```

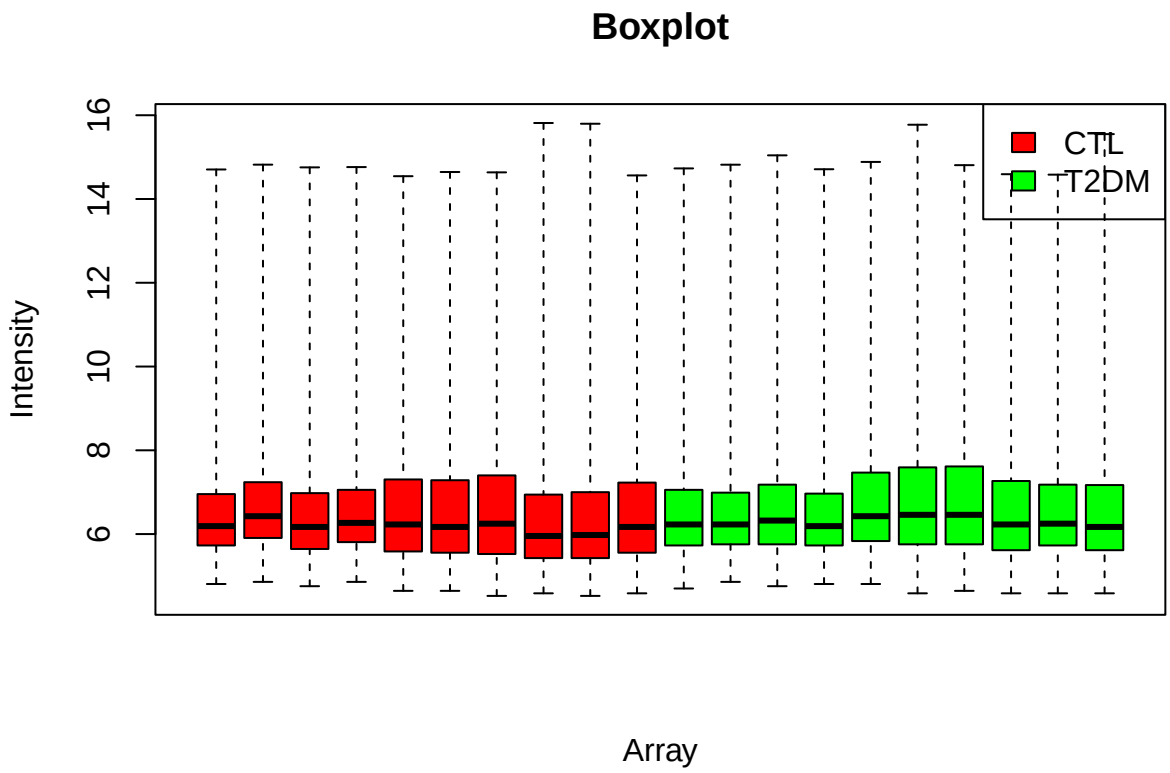
```r
# Align datMeta_T2DM and datExpr_T2DM by sample identifiers
GSM_T2DM <- rownames(pData(data.affy_T2DM))
GSM_T2DM <- substr(GSM_T2DM, 1, 9)
idx_T2DM <- match(GSM_T2DM, datMeta_T2DM$geo_accession)
datMeta_T2DM <- datMeta_T2DM[idx_T2DM, ]
colnames(datExpr_T2DM) <- rownames(datMeta_T2DM)
```

```r
# Cleaning and formatting of GSE20966 metadata
datMeta_T2DM <- datMeta_T2DM[, -c(3:7, 14:36)]
colnames(datMeta_T2DM)[2] <- "Dx"
datMeta_T2DM$Dx[rownames(datMeta_T2DM) %in% c("GSM524151", "GSM524152", "GSM524153","GSM524154", "GSM524
datMeta_T2DM$Dx[rownames(datMeta_T2DM) %in% c("GSM524161", "GSM524162", "GSM524163", "GSM524164", "GSM5
datMeta_T2DM$Dx <- as.factor(datMeta_T2DM$Dx)
```

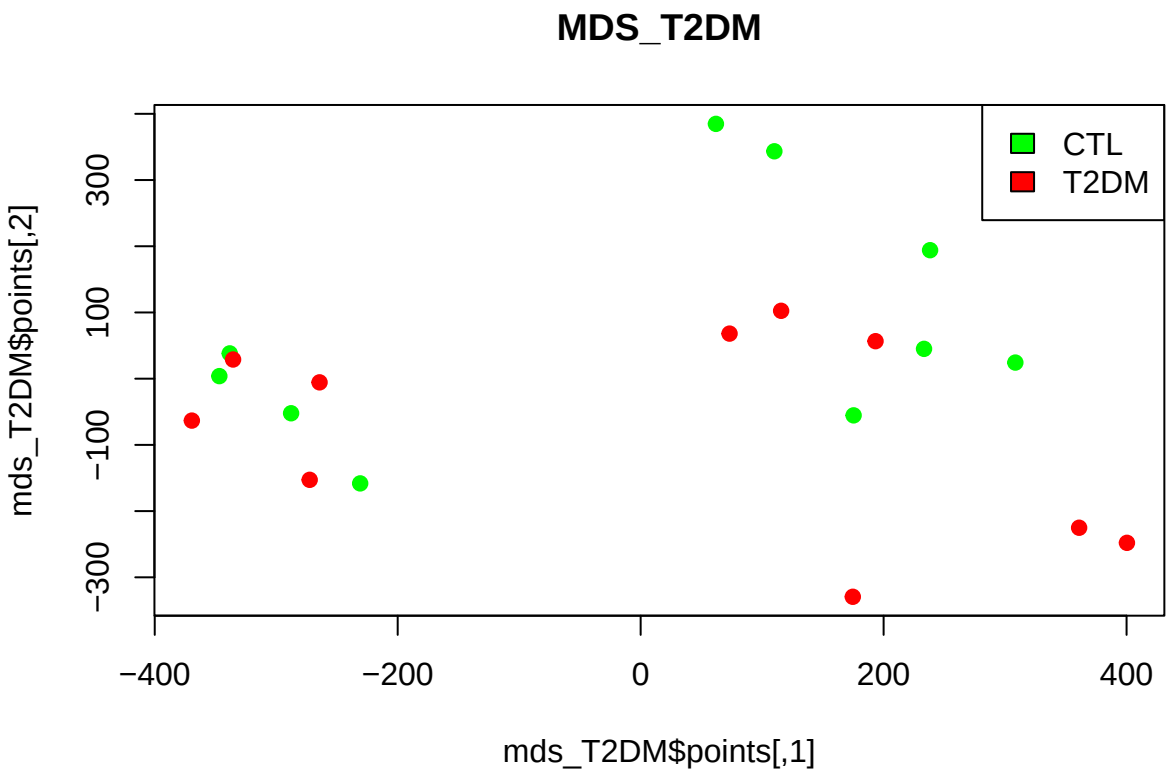
```r
# Preprocessing and quality assessment of GSE20966 raw expression data
datExpr_T2DM <- log2(datExpr_T2DM)
dim(datExpr_T2DM)
```

```
## [1] 1354896      20
```

```r
# Exploratory visualization of GSE20966 raw data
boxplot(datExpr_T2DM,range=0, col=c('red', 'green')[as.numeric(datMeta_T2DM$Dx)], xaxt='n', xlab = "Arra
legend("topright",legend = levels(datMeta_T2DM$Dx),fill = c('red', 'green')[as.numeric(as.factor(levels
```



```
mds_T2DM = cmdscale(dist(t(datExpr_T2DM)),eig=TRUE)
plot(mds_T2DM$points,col=c('green', 'red')[as.numeric(datMeta_T2DM$Dx)],pch=19,main="MDS_T2DM")
legend("topright",legend = levels(datMeta_T2DM$Dx),fill =c('green', 'red')[as.numeric(as.factor(levels(
```




```
# Normalization using RMA
datExpr_T2DM <- rma(data.affy_T2DM, background=T, normalize=T, verbose=T)
```

```
## Warning: replacing previous import 'AnnotationDbi::tail' by 'utils::tail' when
## loading 'u133x3pcdf'
```

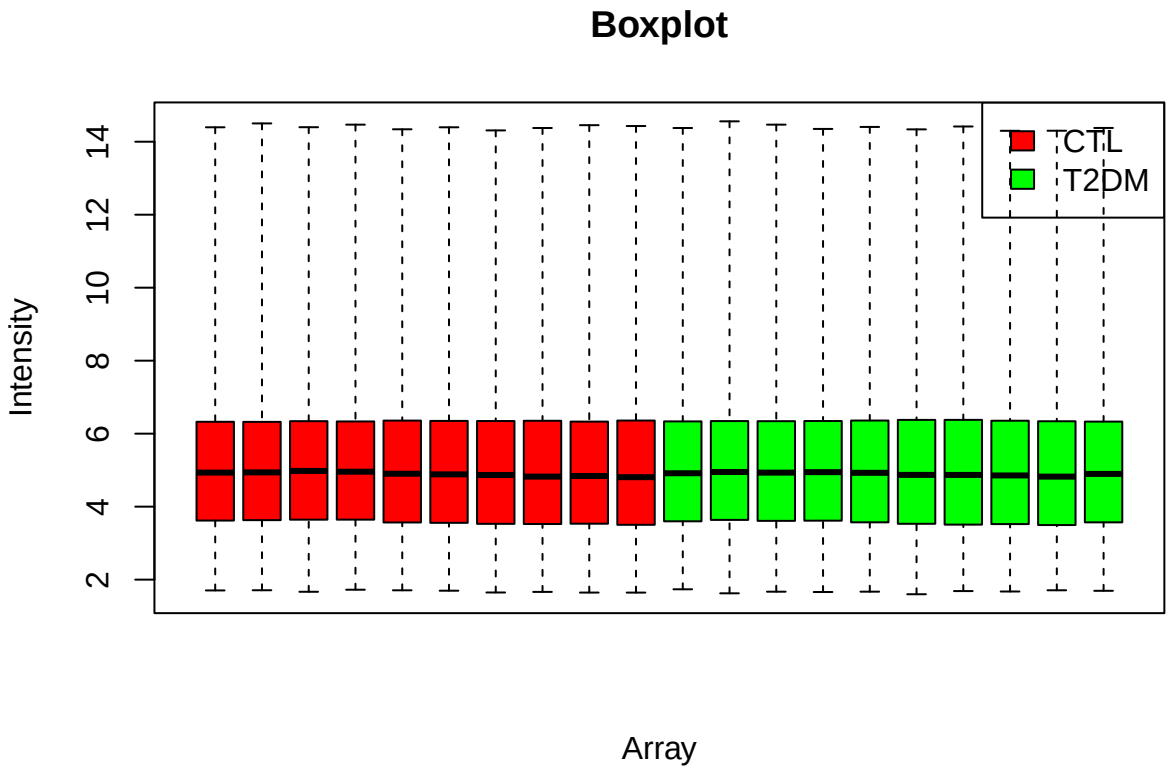
```
## Warning: replacing previous import 'AnnotationDbi::head' by 'utils::head' when
## loading 'u133x3pcdf'
```

```
##
```

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

```
datExpr_T2DM <- exprs(datExpr_T2DM)
```

```r
# Exploratory visualization of GSE20966 normalized data
boxplot(datExpr_T2DM,range=0, col=c('red', 'green')[as.numeric(datMeta_T2DM$Dx)], xaxt='n', xlab = "Arra
legend("topright",legend = levels(datMeta_T2DM$Dx),fill = c('red', 'green')[as.numeric(as.factor(levels
```




```r
# QC analysis with arrayQualityMetrics
datMeta_proc_T2DM <- new("AnnotatedDataFrame", data = datMeta_T2DM)
colnames(datExpr_T2DM) <- rownames(datMeta_T2DM)
eset_T2DM <- new("ExpressionSet", exprs = datExpr_T2DM, phenoData = datMeta_proc_T2DM)

arrayQualityMetrics(expressionset = eset_T2DM,
                    outdir = "/Users/lorandacalderonzamora/Downloads/QC_GSE20966_Report",
                    force = TRUE,
                    do.logtransform = FALSE)
```

```
## The directory '/Users/lorandacalderonzamora/Downloads/QC_GSE20966_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
## 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
## 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
```

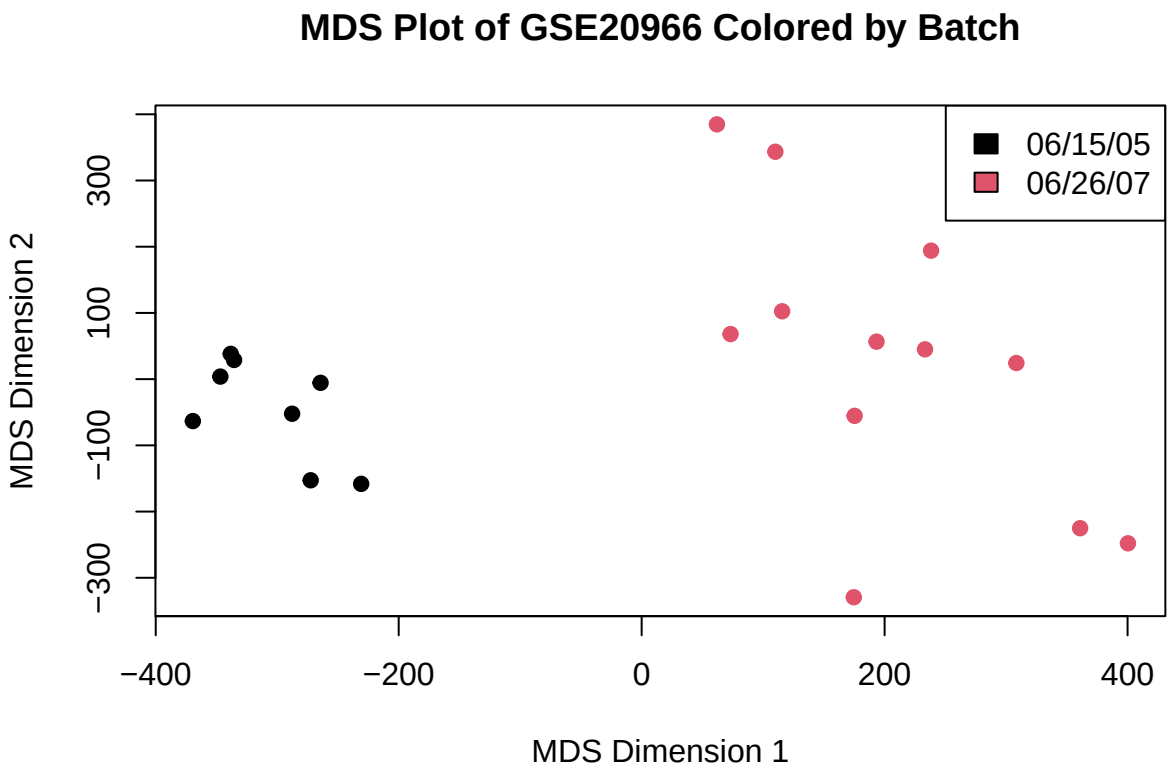
```
## (loaded the KernSmooth namespace)
```

```
# Extract ScanDate from GSE20966 for batch effect correction
batch_T2DM <- protocolData(data.affy_T2DM)$ScanDate
batch_T2DM <- substr(batch_T2DM,1,8)
batch_T2DM <- as.factor(batch_T2DM)
table(batch_T2DM)
```

```
## batch_T2DM
## 06/15/05 06/26/07
##        8       12
```

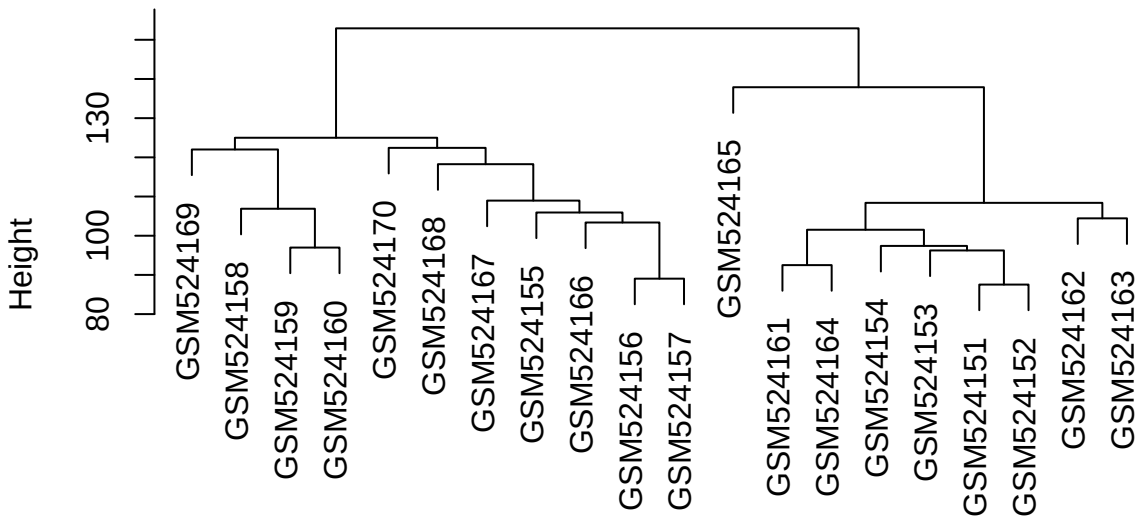
```
datMeta_T2DM$Batch <- batch_T2DM
```

```
# Visualization of ScanDate metadata from GSE20966 to identify potential batch effects
plot(mds_T2DM$points,col = as.numeric(datMeta_T2DM$Batch),pch=19,main="MDS Plot of GSE20966 Colored by B
legend("topright",legend = levels(datMeta_T2DM$Batch),fill = as.numeric(as.factor(levels(datMeta_T2DM$Ba
```

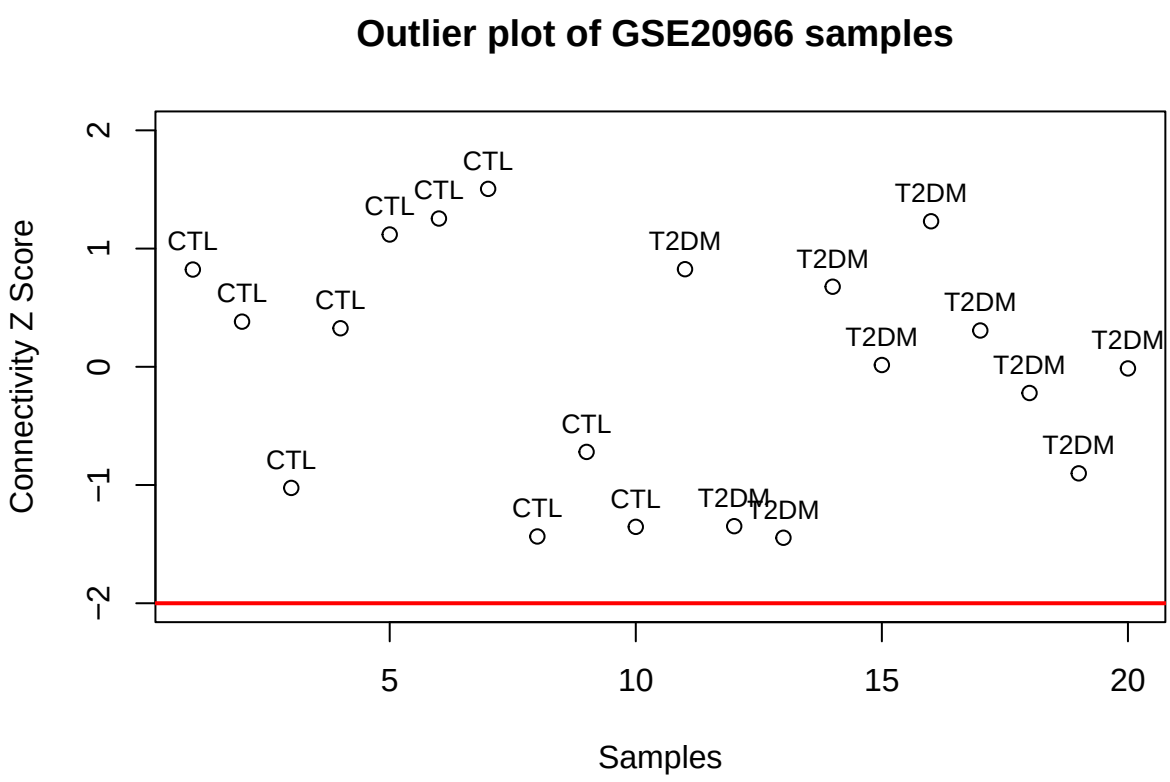



```
# Create ExpressionSet object after Batch effect assessment
datMeta_T2DM$Batch <- batch_T2DM
datMeta_proc_T2DM <- new("AnnotatedDataFrame", data = datMeta_T2DM)
colnames(datExpr_T2DM) <- rownames(datMeta_T2DM)
datAll_T2DM <- new("ExpressionSet", exprs = datExpr_T2DM, phenoData = datMeta_proc_T2DM)
# No singular batch was detected in the GSE20966 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_T2DM <- hclust(dist(t(exprs(datAll_T2DM))), method = "average")
plot(tree_T2DM, main = "Hierarchical clustering of GSE20966 samples", xlab = "", sub = "")
```




```
normadj_T2DM <- (0.5 + 0.5*bicor(exprs(datAll_T2DM)))^2
netsummary_T2DM <- fundamentalNetworkConcepts(normadj_T2DM)
C_T2DM <- netsummary_T2DM$Connectivity
Z.C_T2DM <- (C_T2DM - mean(C_T2DM)) / sqrt(var(C_T2DM))

datLabel_T2DM <- pData(datAll_T2DM)$Dx
plot(1:length(Z.C_T2DM),Z.C_T2DM,main="Outlier plot of GSE20966 samples ",xlab = "Samples",ylab="Connec
text(1:length(Z.C_T2DM),Z.C_T2DM,label=datLabel_T2DM,pos=3,cex=0.8)
abline(h= -2, col="red", lwd = 2)
```




```
# Identify and remove potential outlier from GSE20966 samples based on connectivity Z-score
# No samples exceeded the threshold (Z < -2), so none were removed
to_keep_T2DM <- abs(Z.C_T2DM) < 2
table(to_keep_T2DM)
```

```
## to_keep_T2DM
## TRUE
##   20
```

```
colnames(exprs(datAll_T2DM))[!to_keep_T2DM]
```

```
## character(0)
```

```
datAll_T2DM <- datAll_T2DM[, to_keep_T2DM]
```

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

```
# Annotating Probes for GSE20966 dataset
identifier_T2DM <- "affy_u133_x3p"
getinfo_T2DM <- c("affy_u133_x3p", "ensembl_gene_id", "entrezgene_id", "external_gene_name")
geneDat_T2DM <- getBM(attributes = getinfo_T2DM,
                      filters = identifier_T2DM,
```

```
                      values = rownames(exprs(datAll_T2DM)),
                      mart = ensembl)
idx_T2DM <- match(rownames(exprs(datAll_T2DM)), geneDat_T2DM$affy_u133_x3p)
geneDat_T2DM <- geneDat_T2DM[idx_T2DM, ]
table(is.na(geneDat_T2DM$ensembl_gene_id))
```

```
##
## FALSE  TRUE
## 47996 13363
```

```
to_keep_T2DM <- !is.na(geneDat_T2DM$ensembl_gene_id)
geneDat_T2DM <- geneDat_T2DM[to_keep_T2DM, ]
datAll_T2DM <- datAll_T2DM[to_keep_T2DM, ]
```

```
# Collapse Rows for GSE20966 by Ensembl Gene ID
table(duplicated(geneDat_T2DM$affy_u133_x3p))
```

```
##
## FALSE
## 47996
```

```
table(duplicated(geneDat_T2DM$ensembl_gene_id))
```

```
##
## FALSE  TRUE
## 23693 24303
```

```
CR_T2DM <- collapseRows(exprs(datAll_T2DM),
                        rowGroup = geneDat_T2DM$ensembl_gene_id,
                        rowID = geneDat_T2DM$affy_u133_x3p)
CRdata_T2DM <- CR_T2DM$datETcollapsed
idx_T2DM <- match(CR_T2DM$group2row[,"selectedRowID"], geneDat_T2DM$affy_u133_x3p)
geneDat_T2DM <- geneDat_T2DM[idx_T2DM, ]
rownames(geneDat_T2DM) <- geneDat_T2DM$ensembl_gene_id
```

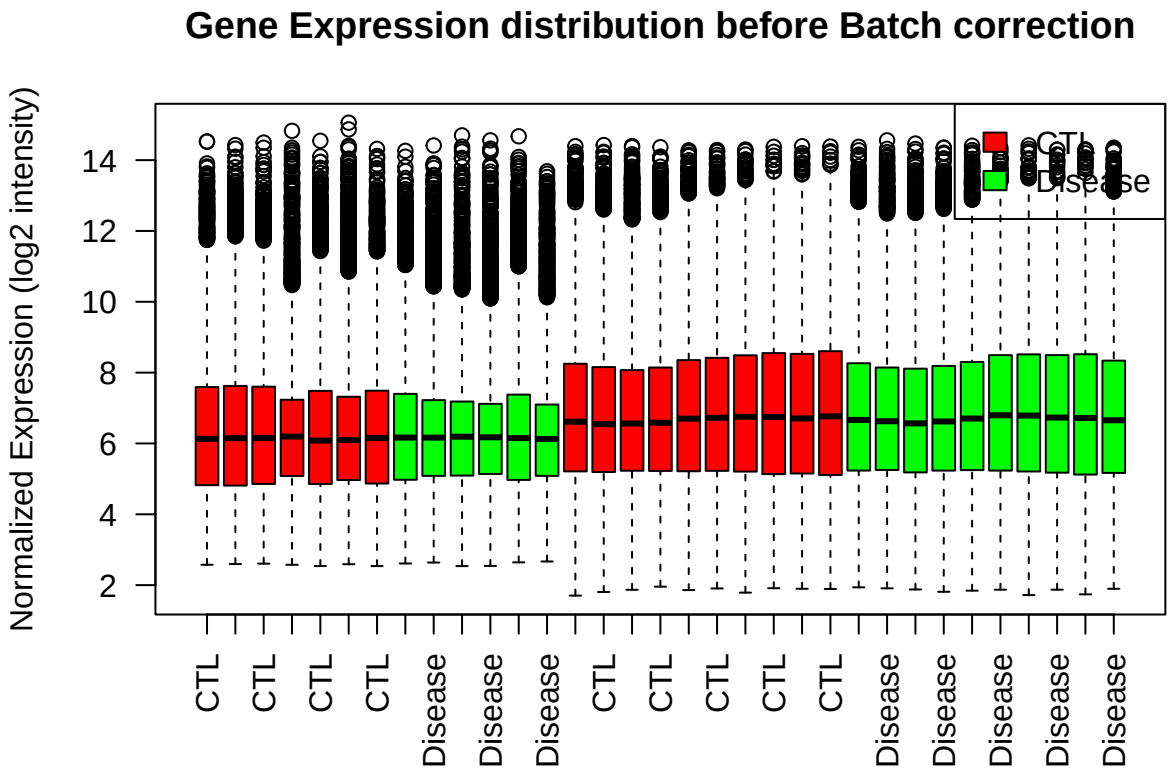
```
# Load a GSE25754 csv file
data_GSE25754 <- read.csv("/Users/lorandacalderonzamora/Downloads/CRdata_T2DM.csv", row.names = 1)
```

```
# Merging GSE20966 and GSE25754 Expression profiles by common genes
common_genes <- base::intersect(rownames(data_GSE25754), rownames(CRdata_T2DM))
data_GSE25754_common <- data_GSE25754[common_genes, ]
CRdata_T2DM_common <- CRdata_T2DM[common_genes, ]

unificated_expr_matrix <- cbind(data_GSE25754_common, CRdata_T2DM_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
```

```r
group_labels[group_labels %in% c("GSM631755", "GSM631756", "GSM631757", "GSM631758", "GSM631759", "GSM63

group_labels[group_labels %in% c("GSM631762", "GSM631763", "GSM631764", "GSM631765", "GSM631766", "GSM63

colnames(unificated_expr_matrix) <- group_labels
```

```r
# Boxplot of the merged Expression matrix GSE20966 and GSE25754 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"))
```



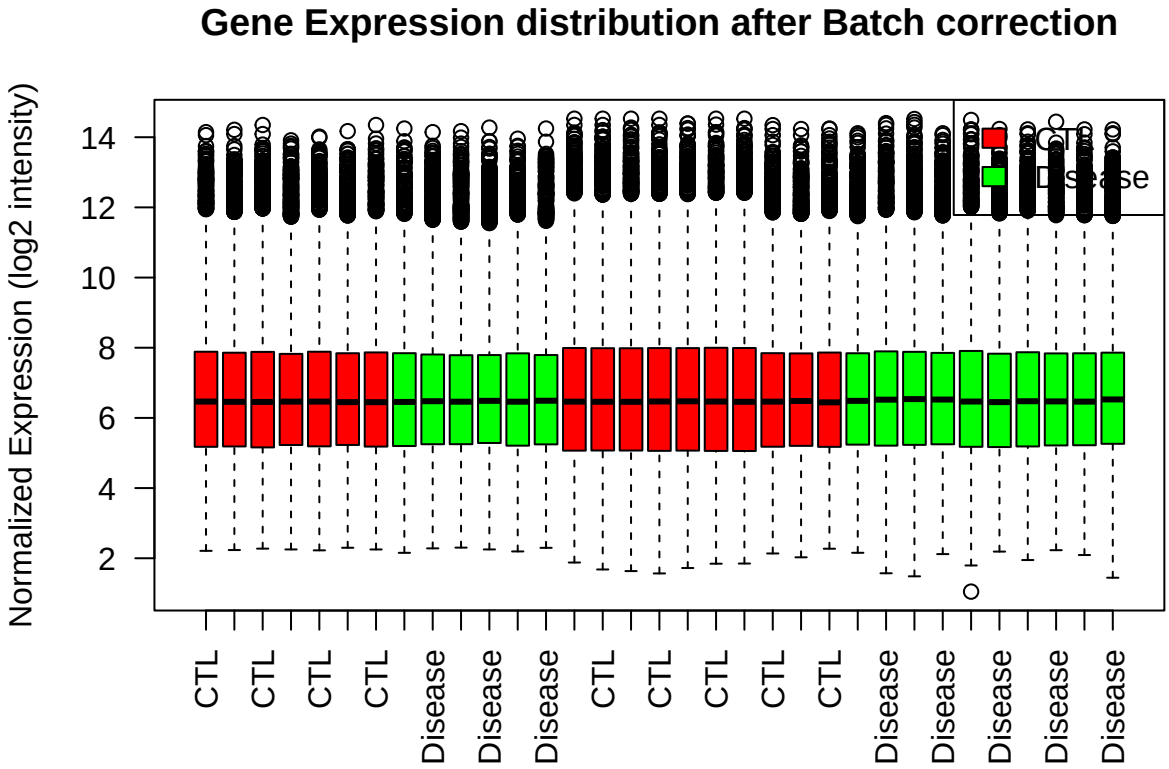

```r
# Quantile normalization of merged Expression matrix
library(preprocessCore)
expr_matrix_qn <- normalize.quantiles(as.matrix(unificated_expr_matrix))
rownames(expr_matrix_qn) <- rownames(unificated_expr_matrix)
colnames(expr_matrix_qn) <- colnames(unificated_expr_matrix)
```

```r
# Batch effect correction between GSE20966 and GSE28345 Datasets using ComBat
n_GSE20966 <- ncol(CRdata_T2DM_common)
n_GSE25754 <- ncol(data_GSE25754_common)
batch <- c(rep("GSE20966", n_GSE20966), rep("GSE25754", n_GSE25754))

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

```
## Found2batches
```

```
## Adjusting for0covariate(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 GSE20966 and GSE25754 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"))
```




```
# 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 GSE20966 and GSE
design <- model.matrix(~ group)
fit_T2DM <- lmFit(combat_expr, design)
fit_T2DM <- eBayes(fit_T2DM)
tt_T2DM <- topTable(fit_T2DM, coef = 2, number = Inf)
head(tt_T2DM)
```

```
##                     logFC  AveExpr         t      P.Value   adj.P.Val        B
## ENSG00000168216 -1.278585 8.713792 -5.860026 1.314637e-06 0.004994375 5.268853
## ENSG00000169062 -1.499784 8.371301 -5.822489 1.470818e-06 0.004994375 5.166687
## ENSG00000143156 -1.326438 7.412049 -5.796852 1.588052e-06 0.004994375 5.096875
## ENSG00000164751 -1.511193 7.316987 -5.794864 1.597525e-06 0.004994375 5.091461
## ENSG00000125827 -1.525400 9.774117 -5.678160 2.265377e-06 0.004994375 4.773339
## ENSG00000115446 -1.175341 8.437421 -5.626459 2.644669e-06 0.004994375 4.632269
```

```r
# Annotating Differential Expression using Ensembl gene IDs
gene_ids <- rownames(tt_T2DM)
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_T2DM$ensembl_gene_id <- rownames(tt_T2DM)
tt_annotated <- merge(tt_T2DM, 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 -0.28388009 7.283546
## 2 ENSG00000000005               TNMD         64102  0.29543124 3.773726
## 3 ENSG00000000419               DPM1          8813 -0.63897214 8.118462
## 4 ENSG00000000457              SCYL3         57147 -0.52107542 7.186853
## 5 ENSG00000000460              FIRRM         55732 -0.01348305 3.337939
## 6 ENSG00000000938                FGR          2268  0.18125732 6.158225
##            t    P.Value  adj.P.Val        B
## 1 -0.9771410 0.33540963 0.48599266 -5.648085
## 2  2.3823258 0.02294733 0.07619349 -3.548506
## 3 -2.0150673 0.05186946 0.12961264 -4.235677
## 4 -1.6386756 0.11051481 0.21959262 -4.845546
## 5 -0.1454852 0.88518796 0.93035350 -6.098616
## 6  1.5866650 0.12185791 0.23509036 -4.921582
```