Microarray Data - GSE5145

General info

This dataset was generated in 2007 and aimed to identify the genes regulated by Vit D in primary bronchial smooth muscle cells. Cells were treated for 24 hours with 100 nM of Vit D3. The experiment was done in triplicate form (a total of 6 samples were analyzed). All samples of this study were used for our data analysis.

Microarray type: Affymetrix Human Genome U133 Plus 2.0 Array [HG-U133_Plus_2]

Preparation

Loading Libraries

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
library(oligo)
```

```
## 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, 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
```

```
## Loading required package: oligoClasses
```

```
## Welcome to oligoClasses version 1.58.0
```

```
## Loading required package: Biobase
```

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

```
## Loading required package: Biostrings
```

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

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

```
##
## Attaching package: 'S4Vectors'
```

```
## 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:grDevices':
##
##     windows
## Loading required package: XVector
## Loading required package: GenomeInfoDb
##
## Attaching package: 'Biostrings'
## The following object is masked from 'package:base':
##
##     strsplit
## ========================================================================================
## Welcome to oligo version 1.60.0
## ========================================================================================
```

```
library(GEOquery)
```

```
## Setting options('download.file.method.GEOquery'='auto')
## Setting options('GEOquery.inmemory.gpl'=FALSE)
```

```
library(arrayQualityMetrics)
library(limma)
```

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

Download and Extract Data

Download and extract microarray .CEL files.

```
url = "https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE5145&format=file"
tf = tempfile()
download.file(url, tf)
untar(tarfile = tf, exdir = "./data/array5145/")
unlink(tf)

l <- list.files(
  "./data/microarray/GSE5145", pattern = ".+\\.gz",  full.names = TRUE)
print(l)
lapply(l, function(x) { gunzip(x, overwrite = TRUE) } )
```

Download and extract annotation file.

```
url = "https://ftp.ncbi.nlm.nih.gov/geo/series/GSE5nnn/GSE5145/soft/GSE5145_family.soft.gz"
download.file(url, "./data/annotation/GSE5145/GSE5145_family.soft.gz")

l <- list.files(
  "./data/annotation/GSE5145", pattern = ".+\\.gz",  full.names = TRUE)
print(l)
lapply(l, function(x) { gunzip(x, overwrite=TRUE) } )
```

Load the Data

The .CEL files are loaded into an array object.

```
exonCELs <- list.celfiles("./data/array5145/")
arrayRaw <- read.celfiles(
  paste(rep("./data/array5145/", length(exonCELs)), exonCELs, sep = ""))
```

The differential expression between Vit D treated and untreated primary bronchial smooth muscle cell samples will be investigated. According to the annotation provided, no confounding factors are present. All cells were derived from the same tissue sample of a 31 year old Hispanic male.

```
annot <- getGEO(filename = "./data/annotation/GSE5145/GSE5145_family.soft")

IDs <- annot@header$sample_id
titles <- c()

for (id in IDs) {
  title <- annot@gsms[[id]]@header[["title"]]
  titles <- append(titles, title)
}
data.frame(IDs, titles)
```

```
##         IDs               titles
## 1 GSM116101 hBSMC_vitamin D_rep1
## 2 GSM116102 hBSMC_vitamin D_rep2
## 3 GSM116103 hBSMC_vitamin D_rep3
## 4 GSM116104   hBSMC_control_rep1
## 5 GSM116105   hBSMC_control_rep2
## 6 GSM116106   hBSMC_control_rep3
```

Data exploration

The dimensions and a general overview of the data is generated. The first three `.CEL` files correspond to the Vit D deficient samples, the last three are the control samples.

```
print(
  paste(
    "Array matrix dimensions:",
    dim(exprs(arrayRaw))[1], 'x',
    dim(exprs(arrayRaw))[2], sep = ' '))
```

```
## [1] "Array matrix dimensions: 1354896 x 6"
```

```
data.frame(head(exprs(arrayRaw)))
```

```
##   GSM116101.CEL GSM116102.CEL GSM116103.CEL GSM116104.CEL GSM116105.CEL
## 1            92            90            86            75            74
## 2         10958         10352          9758          9714          9617
```

```
## 3           110            91           108           110           127
## 4         11088         10776         10046         10016          9818
## 5           129            85           113            80            91
## 6            80            71            71            74            58
##   GSM116106.CEL
## 1            97
## 2         10725
## 3           117
## 4         11283
## 5            77
## 6            89
```

Data Extraction

Expression values are extracted using the robust multichip average method (RMA) and prode IDs are mapped to their corresponding probe set.

```
arrayRma = rma(arrayRaw)
```

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

```
print(
  paste(
    "Array matrix dimensions after summarization:",
    dim(exprs(arrayRma))[1], 'x',
    dim(exprs(arrayRma))[2], sep = ' '))
```

```
## [1] "Array matrix dimensions after summarization: 54675 x 6"
```

```
data.frame(head(exprs(arrayRma)))
```

```
##           GSM116101.CEL GSM116102.CEL GSM116103.CEL GSM116104.CEL GSM116105.CEL
## 1007_s_at      7.361670      7.242969      7.194826      7.545413      7.527583
## 1053_at        4.943707      4.975347      4.841540      5.050087      4.937637
## 117_at         3.673319      3.613580      3.652913      3.865110      3.589851
## 121_at         6.330822      6.438481      6.173505      6.261498      6.296260
## 1255_g_at      2.466963      2.391853      2.263404      2.404987      2.264599
## 1294_at        6.099451      6.294354      6.237696      6.601863      6.523832
##           GSM116106.CEL
## 1007_s_at      7.498058
## 1053_at        5.085002
## 117_at         3.745871
## 121_at         6.458236
## 1255_g_at      2.504713
## 1294_at        6.467871
```

All probes were mapped to unique genes.

```
sum(duplicated(rownames(arrayRma)))
```

```
## [1] 0
```

Quality Control

Quality of the array is assessed using the `arrayQualityMetrics` package. From this analysis we can conclude that the overall quality of the arrays is quite good. However, sample 6 shows some signs of being an outlier,

and the graph on figure 7 (Standard deviation versus rank of the mean) shows a slight upward trend, which is symptomatic of a saturation of the intensities.

```
arrayQualityMetrics(expressionset = arrayRma,
                    outdir = "./stats/Microarray/GSE5145/",
                    force = TRUE)
```

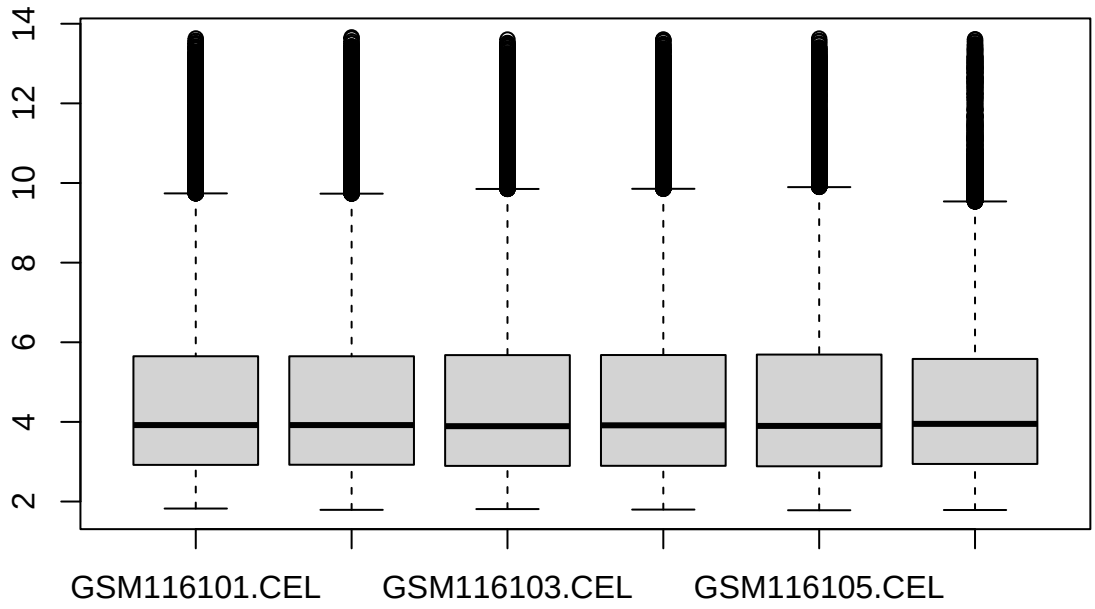
```
## The report will be written into directory './stats/Microarray/GSE5145/'.
```

```
## 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)
```

The data distributions also conform to these findings.

```
boxplot(exprs(arrayRma))
```



Differential Expression Analysis

Model Fitting

LIMMA is performed to find differentially expressed genes.

```
# design matrix
design <- model.matrix(~ factor(c(1, 1, 1, 0, 0, 0)))
colnames(design) <- c("Intercept", "Treatment")
fit <- lmFit(arrayRma, design)
```

To make a pair-wise comparison between the two groups the appropriate contrast matrix is created.

```
contrast.matrix <- makeContrasts(Treatment, levels = design)
fit2 <- contrasts.fit(fit, contrast.matrix)
fit2 <- eBayes(fit2)
```

Identifying Top Differentially Expressed Genes

A list of top genes differential expressed in the Control versus treatment group can now be obtained.

The best probe (203887_s_at) is expressed 10^3 times more in the treated group (log fold change).

```
topNumber <- 1000
table <- topTable(
  fit2,
  coef = 1,
  number = topNumber,
  adjust = "BH") # FDR correction of p-values
```

```
sig_table <- table[table["adj.P.Val"] < 0.05, ]

print(paste("Significant genesets after FDR correction:", dim(sig_table)[1]))
```

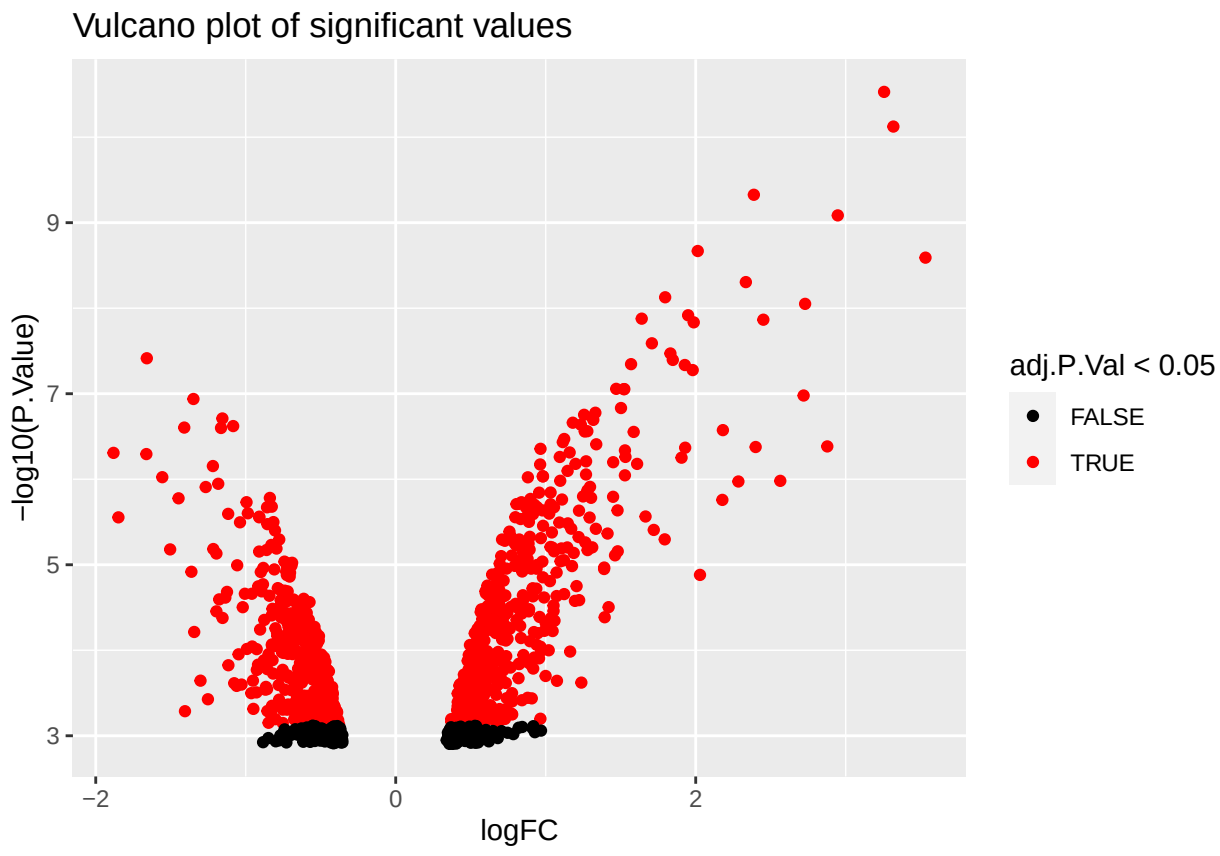
```
## [1] "Significant genesets after FDR correction: 826"
```

```
head(sig_table)
```

```
##                logFC  AveExpr        t      P.Value    adj.P.Val        B
## 203887_s_at 3.256129 7.314559 42.79857 2.955288e-11 1.615804e-06 12.24852
## 223484_at   3.317873 4.384602 38.34302 7.514957e-11 2.054401e-06 11.97117
## 225685_at   2.387715 7.496611 30.87271 4.714531e-10 8.592233e-06 11.28555
## 203888_at   2.947261 6.710770 28.90961 8.217291e-10 1.123201e-05 11.03823
## 205343_at   2.013635 3.990791 25.80503 2.144143e-09 2.335783e-05 10.56511
## 211470_s_at 3.531888 4.986945 25.26424 2.563274e-09 2.335783e-05 10.47047
```

The vulcano plot shows that significant differentially expressed genes are primarily overexpressed.

```
ggplot(table, aes(x = logFC, y = -log10(P.Value), color = adj.P.Val < 0.05)) +
  geom_point() +
  scale_color_manual(values = c("black","red")) +
  ggtitle("Vulcano plot of significant values")
```



Summarization at the gene level

The probe sets are converted to the gene level. Many genes are only represented by a single probe set, but for some genes multiple probe sets are present

```
# input data (all of equal length)
allProbeIDs <- annot@gpls[["GPL570"]]@dataTable@table[["ID"]]
allGeneIDs <- annot@gpls[["GPL570"]]@dataTable@table[["ENTREZ_GENE_ID"]]
allGeneSym <- annot@gpls[["GPL570"]]@dataTable@table[["Gene Symbol"]]
allGeneOnt <-
  annot@gpls[["GPL570"]]@dataTable@table[["Gene Ontology Biological Process"]]
```

```
# desired data
geneIDs <- c()
geneSym <- c()
logFCs <- c()
geneOnt <- c()

for (topIndex in 1:topNumber) {
  probeID <- rownames(sig_table)[topIndex]
  annotIndex <- grep(probeID, allProbeIDs)

  if (!(allGeneIDs[annotIndex] %in% geneIDs) && (probeID %in% allProbeIDs)) {
    geneIDs <- append(geneIDs, allGeneIDs[annotIndex[1]])
    geneSym <- append(geneSym, allGeneSym[annotIndex[1]])
    logFCs <- append(logFCs, table[[1]][topIndex])
    geneOnt <- append(geneOnt, allGeneOnt[annotIndex[1]])
  }
}

df <- data.frame(
  Entrez.gene.ID = geneIDs,
  Gene.symbol = geneSym,
  Fold.Change = logFCs,
  Gene.ontology = geneOnt)

print(paste("Number of significant genes:", dim(df)[1], sep = ' '))
```

```
## [1] "Number of significant genes: 534"
```

```
head(df)
```

```
##   Entrez.gene.ID Gene.symbol Fold.Change
## 1           7056        THBD    3.256129
## 2          84419    C15orf48    3.317873
## 3          10602    CDC42EP3    2.387715
## 4           6819     SULT1C2    2.013635
## 5          54518     APBB1IP    1.796377
## 6          50486        G0S2    2.729367
##
## 1 0007165 // signal transduction // inferred from electronic annotation /// 0007565 // female pregnar
## 2
## 3
## 4
## 5
## 6
```

The significant probe sets could be mapped to 534 individual genes, which were mostly upregulated after Vit D treatment. Many of these genes have a role in high-level signal transduction (cell cyle, metabolism, differentiation). This is a first indication that Vit D is an important nutrient with pleiotropic regulatory

effects in the cell.

When comparing these findings to those reported in the original paper, the same upregulated genes were found.

Gene Set Analysis

The genes are grouped into gene sets to identify pathways regulated by Vit D.

```
# filter out NA values
res <- kegga(de = df$Entrez.gene.ID,
             universe = allGeneIDs,
                 species.KEGG = "hsa")
res <- res[order(res$N),]

res$FDR.DE <- p.adjust(res$P.DE, n = nrow(res), method = "BH")

print(paste("Number of significant gene sets:", dim(res)[1], sep = ' '))
```

```
## [1] "Number of significant gene sets: 353"
```

```
print(head(res[order(res$DE, decreasing = TRUE),]))
```

```
##                                       Pathway    N DE         P.DE       FDR.DE
## path:hsa01100              Metabolic pathways 1434 46 3.535156e-02 1.402146e-01
## path:hsa05200              Pathways in cancer  515 27 1.815954e-04 7.122577e-03
## path:hsa05205        Proteoglycans in cancer  196 20 7.794010e-08 2.751285e-05
## path:hsa04010         MAPK signaling pathway  282 17 6.083896e-04 1.499479e-02
## path:hsa04151     PI3K-Akt signaling pathway  341 17 4.530867e-03 4.569703e-02
## path:hsa05165 Human papillomavirus infection  319 16 5.470081e-03 5.067944e-02
```

Save Results

The resulting genes and gene sets are saved for further comparison with the other omics methods.

```
# saving significant genes
write.csv(df, file = "./results/GSE5145/array5145_genes.csv")

# saving significant gene sets
write.csv(res, file = "./results/GSE5145/array5145_genesets.csv")
```