Visualization, Differential Expression Analysis and Downstream Analysis of Glioma Transcriptomics Count Data

Nishat Tamanna

2024-09-10

The whole analysis will use a count dataset of glioblastoma transcriptomic samples. This dataset contains 5 samples from Recurrent Tumor and 5 from Primary Tumor.

Installing gplots for visualization

```
install.packages("gplots")
library("gplots")
```

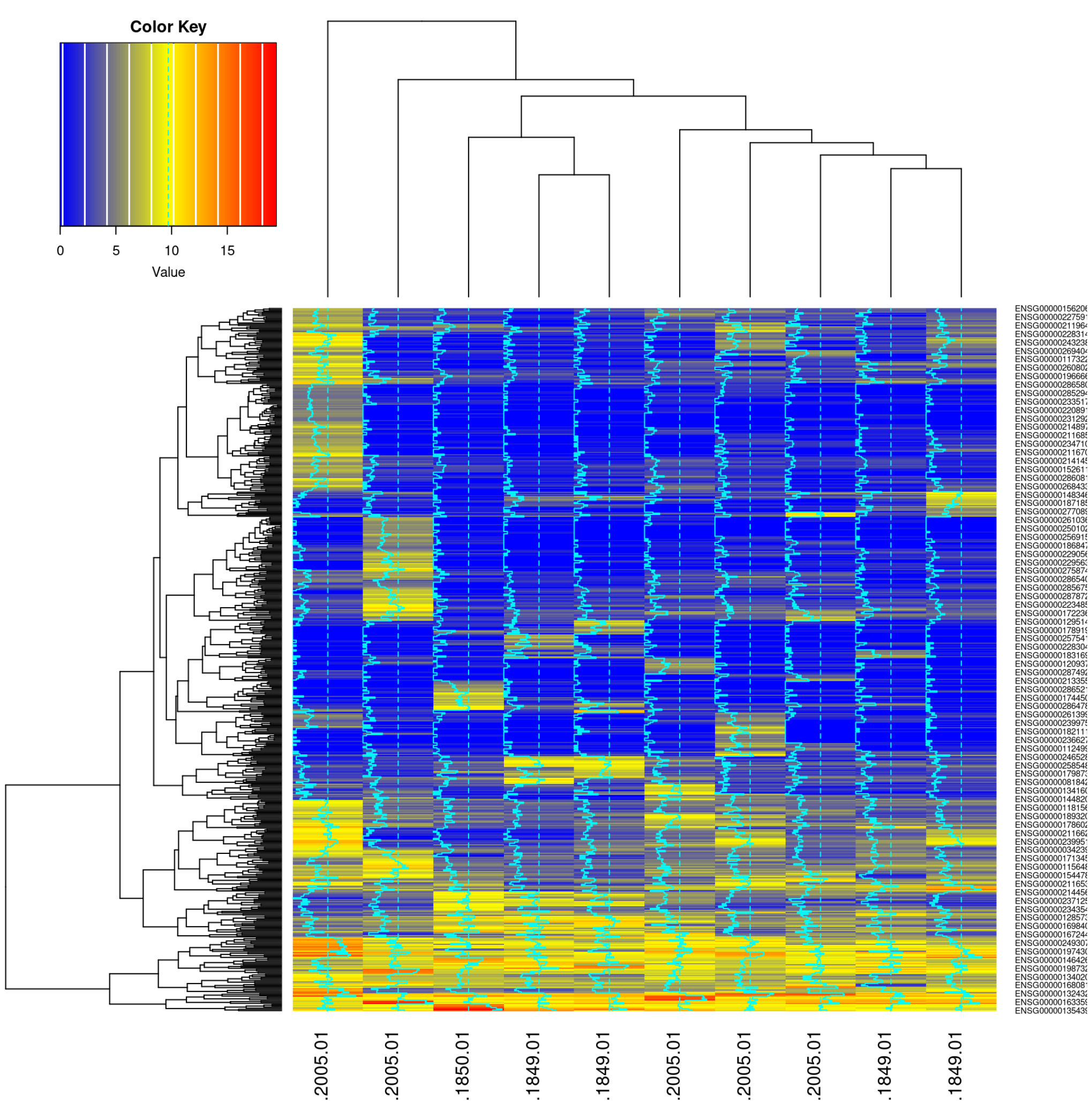
Generating Matrix from a CSV file

```
raw_counts <- read.csv('/home/hp/Documents/HackbioCancer/glioblastoma_count_file.csv', row.names=1)
mat <- as.matrix(raw_counts)
log_data_matrix <- log2(mat+1) #log transformation of matrix as data range is too broad
head(log_data_matrix)
```

```
##                 TCGA.19.4065.02A.11R.2005.01 TCGA.19.0957.02A.11R.2005.01
## ENSG00000272398                     9.577429                    12.144340
## ENSG00000135439                    11.430453                    13.033595
## ENSG00000130348                     9.876517                     9.733015
## ENSG00000198719                     7.857981                    10.307201
## ENSG00000169429                     9.079485                     9.002815
## ENSG00000171608                    10.325305                     9.493855
##                 TCGA.06.0152.02A.01R.2005.01 TCGA.14.1402.02A.01R.2005.01
## ENSG00000272398                     9.417853                    10.830515
## ENSG00000135439                    11.432542                     8.204571
## ENSG00000130348                    10.288866                    10.450180
## ENSG00000198719                     9.675957                     8.845490
## ENSG00000169429                     9.357552                    11.497852
## ENSG00000171608                    10.727070                     8.049849
##                 TCGA.14.0736.02A.01R.2005.01 TCGA.06.5410.01A.01R.1849.01
## ENSG00000272398                    11.604553                     8.154818
## ENSG00000135439                     9.483816                     9.679480
## ENSG00000130348                     9.022368                     9.941048
## ENSG00000198719                     6.845490                     8.257388
## ENSG00000169429                    10.222795                    15.353905
## ENSG00000171608                     8.422065                    11.159241
##                 TCGA.19.5960.01A.11R.1850.01 TCGA.14.0781.01B.01R.1849.01
## ENSG00000272398                    18.906104                     9.891784
## ENSG00000135439                    16.349575                     9.592457
## ENSG00000130348                    15.855793                    10.051209
## ENSG00000198719                    17.335643                     8.400879
## ENSG00000169429                     7.087463                    12.823566
## ENSG00000171608                     8.693487                    10.207014
##                 TCGA.02.2483.01A.01R.1849.01 TCGA.06.2570.01A.01R.1849.01
## ENSG00000272398                     14.45128                     13.28135
## ENSG00000135439                     10.55555                     14.38256
## ENSG00000130348                     10.60177                     10.74567
## ENSG00000198719                     12.40008                     11.99081
## ENSG00000169429                     10.90087                     11.86766
## ENSG00000171608                     14.95233                     10.21796
```

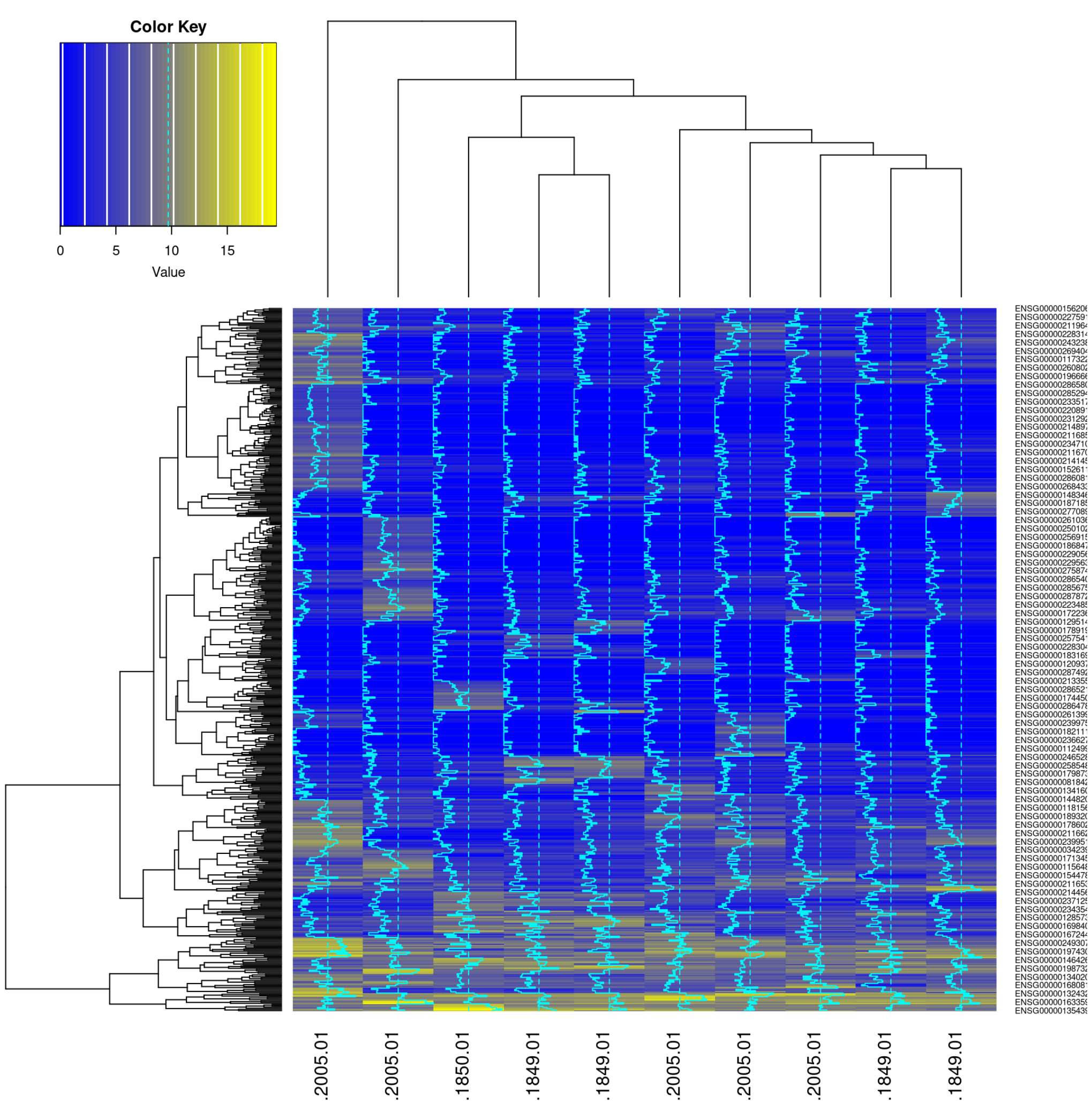
Heatmap using diverging color palette

```
col_palette <- colorRampPalette(c('blue','yellow','red'))(n=300)
heatmap.2(x=log_data_matrix, col = col_palette, density.info = 'none')
```



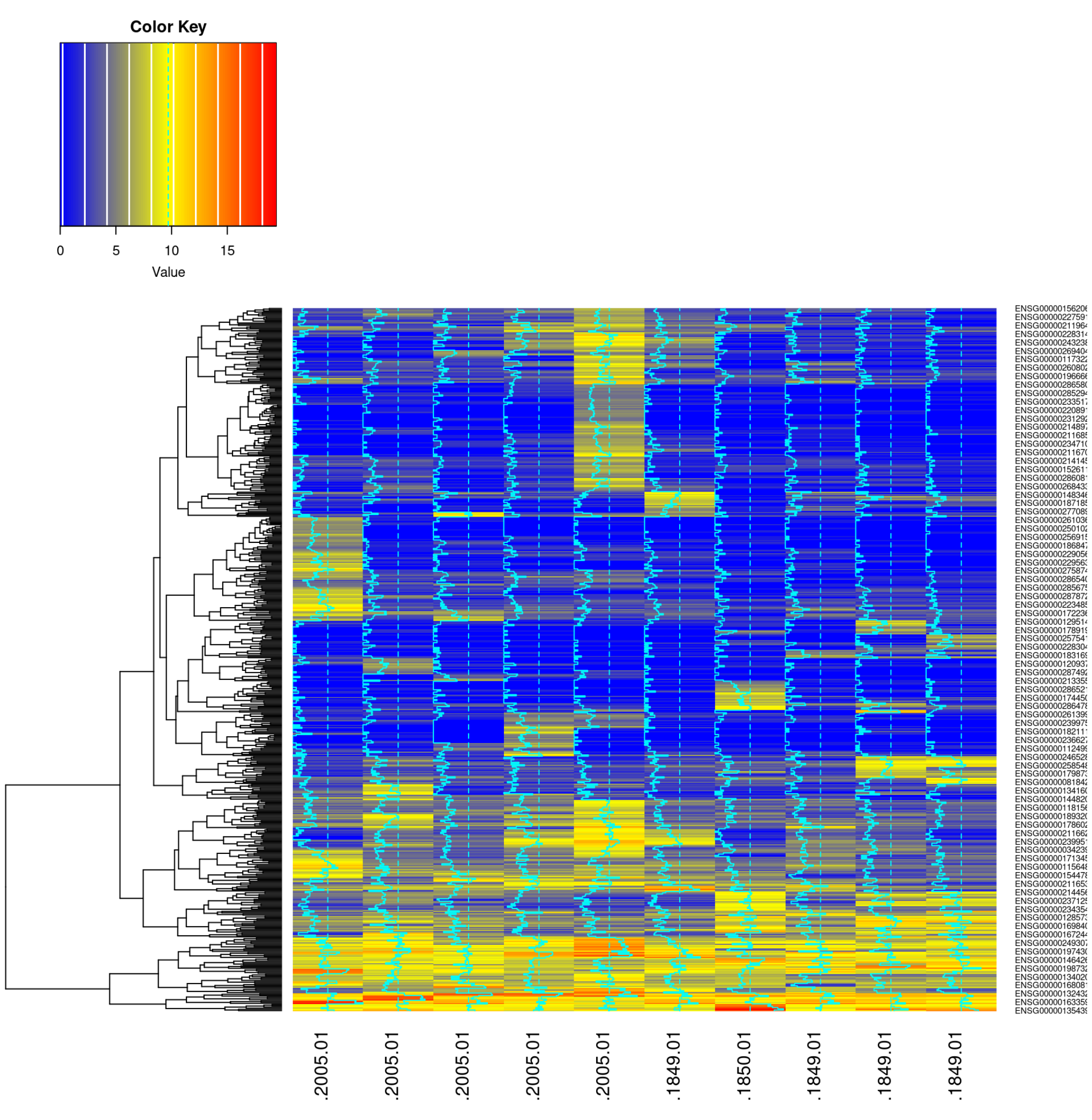
Heatmap using sequential color palette

```
col_palette <- colorRampPalette(c('blue','yellow'))(n=300)
heatmap.2(x=log_data_matrix, col = col_palette, density.info = 'none')
```



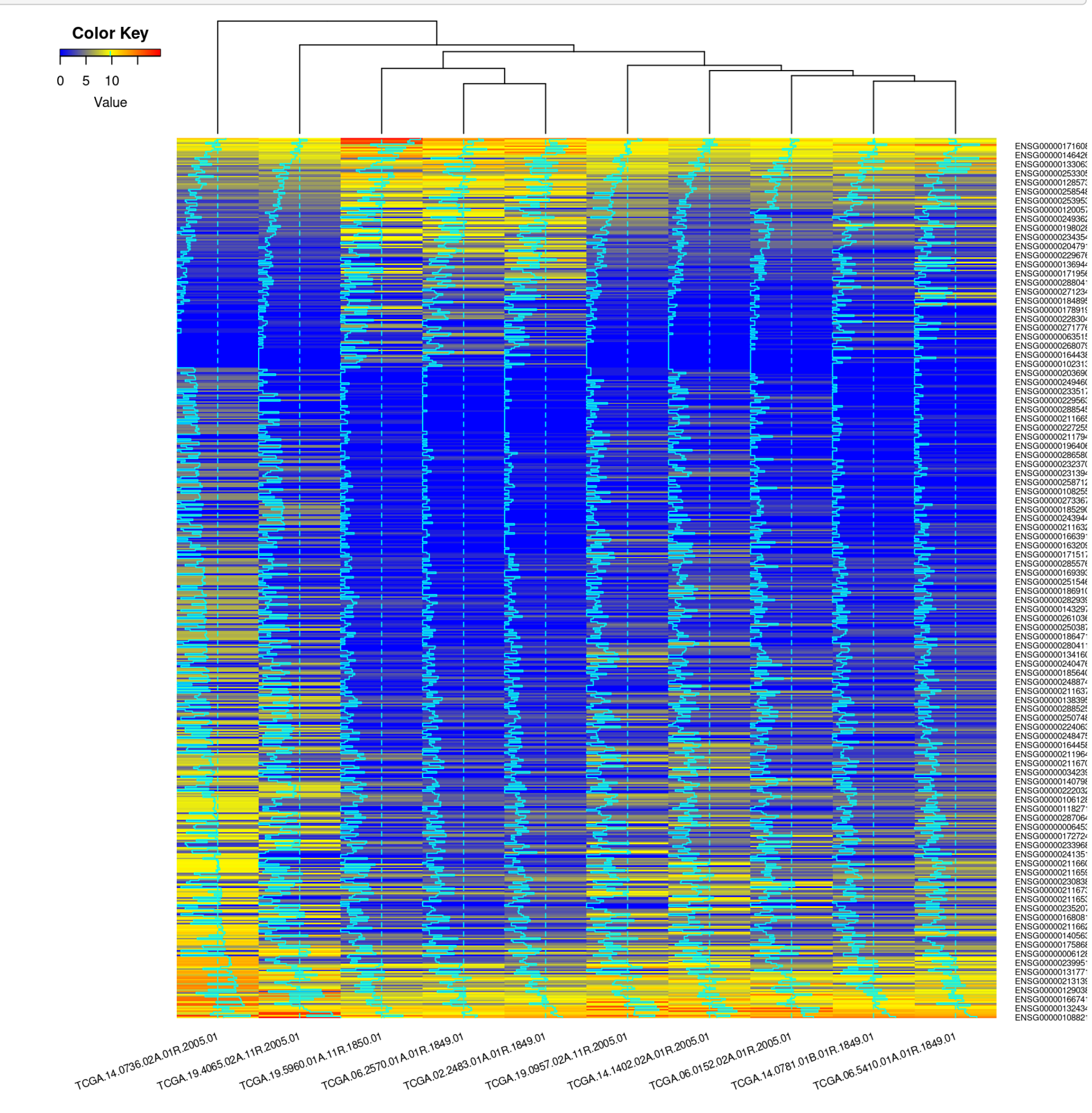
Clustering Genes (rows) Alone

```
col_palette <- colorRampPalette(c('blue','yellow','red'))(n=300)
heatmap.2(x=log_data_matrix, col = col_palette, density.info = 'none', dendrogram = 'row', Rowv = TRUE, Colv = FA
LSE)
```



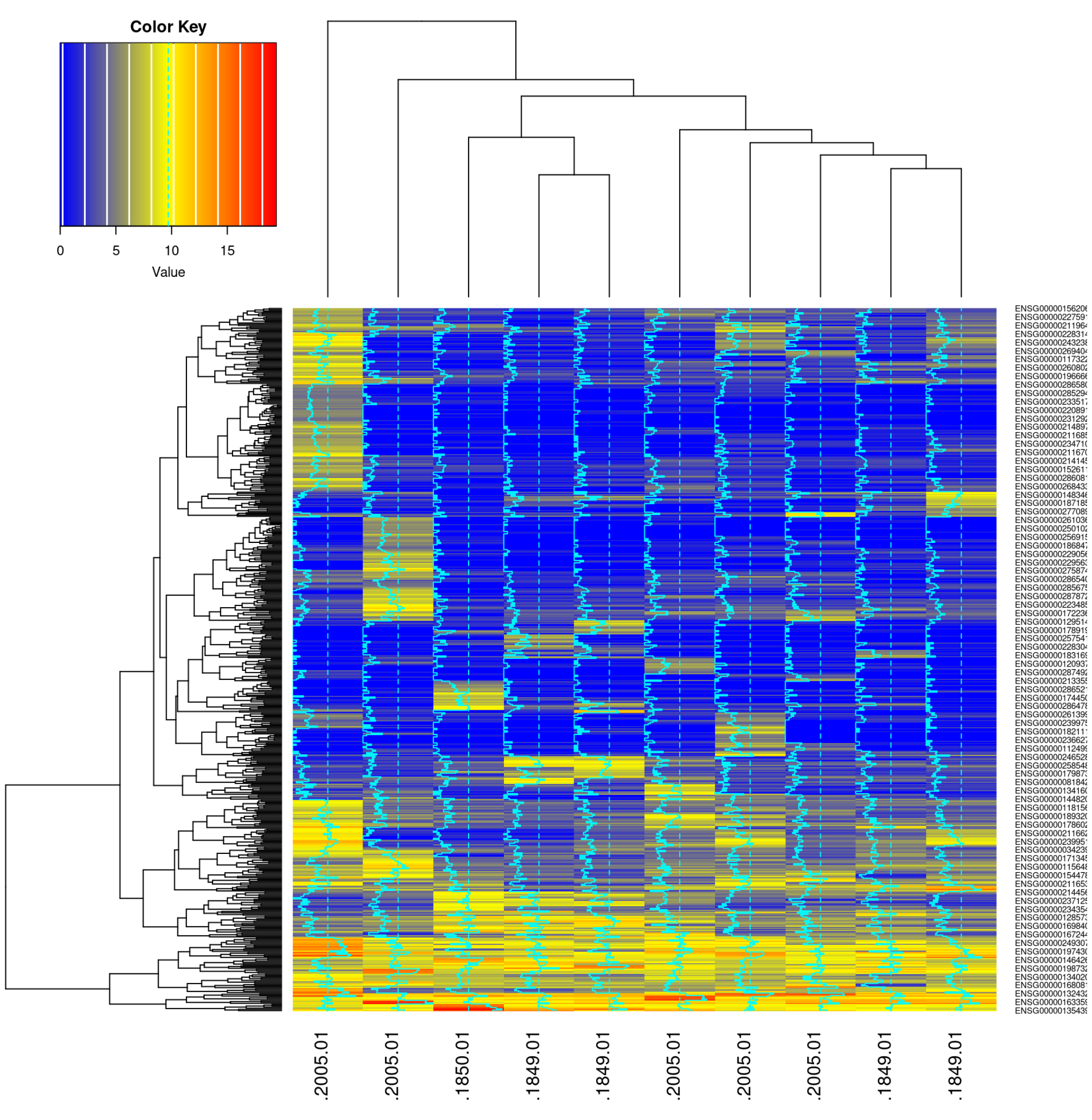
Clustering Samples (columns) Alone

```
par(mar = c(10, 10, 10, 10))
col_palette <- colorRampPalette(c('blue','yellow','red'))(n=300)
heatmap.2(x=log_data_matrix, col = col_palette, density.info = 'none', dendrogram = 'column', Rowv = FALSE, Colv
= TRUE, lmat = rbind(c(4, 3), c(2, 1)),  # Layout matrix
          lhei = c(1, 8),                # Increase the height of the heatmap body
          lwid = c(0.2, 1),                # Adjust widths for the heatmap and key
          cexCol = 0.7, srtCol = 20)
```



Clustering both Genes and Samples

```
col_palette <- colorRampPalette(c('blue','yellow','red'))(n=300)
heatmap.2(x=log_data_matrix, col = col_palette, density.info = 'none', Rowv = TRUE, Colv = TRUE)
```



Installing DeSeq2 package for differential expression analysis

```
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("DESeq2")
library("DESeq2")
```

Preparation for Differential Expression Analysis

```
#Creating Sample Table
sample_table <- data.frame(
  sampleName <- colnames(raw_counts),
  condition = c("Recurret Tumor","Recurret Tumor","Recurret Tumor","Recurret Tumor","Recurret Tumor","Primary Tum
or","Primary Tumor","Primary Tumor","Primary Tumor","Primary Tumor")
)
sample_table$condition <- as.factor(sample_table$condition)
head(sample_table)
```

```
##   sampleName....colnames.raw_counts.      condition
## 1       TCGA.19.4065.02A.11R.2005.01 Recurret Tumor
## 2       TCGA.19.0957.02A.11R.2005.01 Recurret Tumor
## 3       TCGA.06.0152.02A.01R.2005.01 Recurret Tumor
## 4       TCGA.14.1402.02A.01R.2005.01 Recurret Tumor
## 5       TCGA.14.0736.02A.01R.2005.01 Recurret Tumor
## 6       TCGA.06.5410.01A.01R.1849.01  Primary Tumor
```

Creating DESeq Dataset and Running Differential Expression Analysis

```
dds <- DESeqDataSetFromMatrix(countData = raw_counts, colData = sample_table, design = ~ condition)
dds <- DESeq(dds)
```

Filtering out Significant Genes (padj <0.05 and fold change cutoff 2)

```
results_table <- results(dds)
significant_genes <- subset(results_table, padj < 0.05)
significant_genes_up <- subset(results_table, padj < 0.05 & log2FoldChange > 1)
significant_genes_down <- subset(results_table, padj < 0.05 & log2FoldChange < -1)
```

Exporting genes, fold change and adjacent p values to CSV file

```
background_genes <- data.frame(gene = rownames(results_table), log2FC = results_table$log2FoldChange, pval = resu
lts_table$pvalue)
DE_significantly_up_genes <- data.frame(gene = rownames(significant_genes_up), log2FC = significant_genes_up$log2
FoldChange, pval = significant_genes_up$pvalue)
DE_significantly_down_genes <- data.frame(gene = rownames(significant_genes_down), log2FC = significant_genes_dow
n$log2FoldChange, pval = significant_genes_down$pvalue)
head(DE_significantly_up_genes)
```

```
##              gene   log2FC        pval
## 1 ENSG00000202111 4.474860 0.009668808
## 2 ENSG00000118231 3.852969 0.002850270
## 3 ENSG00000286404 3.941243 0.011999617
## 4 ENSG00000287872 3.580704 0.009825419
## 5 ENSG00000288525 5.078708 0.003843982
## 6 ENSG00000259518 5.086102 0.002023824
```

```
# Exporting into a CSV file
write.csv(background_genes, "background_genes.csv", row.names = FALSE)
write.csv(DE_significantly_up_genes, "DE_significantly_up_genes.csv", row.names = FALSE)
write.csv(DE_significantly_down_genes, "DE_significantly_down_genes.csv", row.names = FALSE)
```

Functional Enrichment Analysis with ShinyGO

Using the default parameters in ShinyGO 0.80, for 89 significantly differential expressed genes, we found the following top 10 enriched KEGG pathways.