# Profiling R Codes

## Introduction

Profiling involves running many lines of code to find *bottlenecks*. Sometimes we have a good hunch on which lines of codes are taking up the bulk of the runtime. Other times, we don't. Profiling allows us to pinpoint which lines of codes are bottlenecks with certainty. Hence, allowing us to focus on these bottlenecks and potentially find alternatives to increase overall efficiency. We will be using the `profvis()` function from its eponymous package.

```
install.packages("profvis")
```

## Dataset

In this tutorial, we will be using the GENCODE gene transfer file (GTF) file. A GTF file contains the comprehensive gene, transcript, and exon annotations for a give species. The latest version of a GTF file can be retrieved from the GENCODE repository (https://www.gencodegenes.org/). Here, we will using the human GTF file version 31. The data frame consists of 9 columns and a brief explanation of each of these columns as follows:

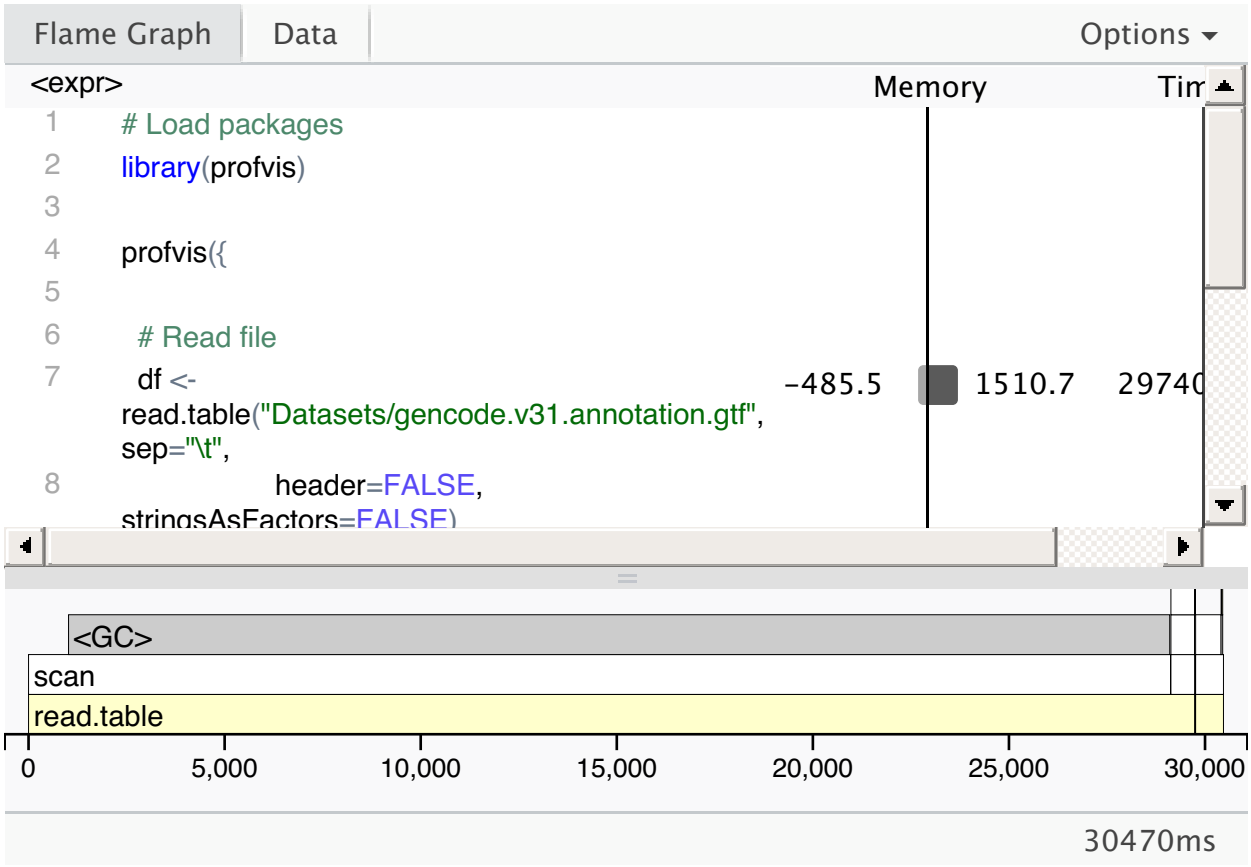
```
## Warning: package 'knitr' was built under R version 3.6.2
```

| Column | Content | Value |
|---|---|---|
| 1 | Chromosome name | 1, 2, 3 |
| 2 | Annotation source | ENSEMBLE, HAVANA |
| 3 | Feature type | gene, transcript, exon |
| 4 | Genomic start location | interger |
| 5 | Genomic end location | interger |
| 6 | Score (not used) | . |
| 7 | Genomic strand | +, - |
| 8 | Genomic phase (for CDS features) | ., 0, 1, 2 |
| 9 | Attributes | gene_id, transcript_id, gene_type, gene_status, gene_name, transcript_type, transcript_status, transcript_name, exon_number, exon_id |

## Profiling

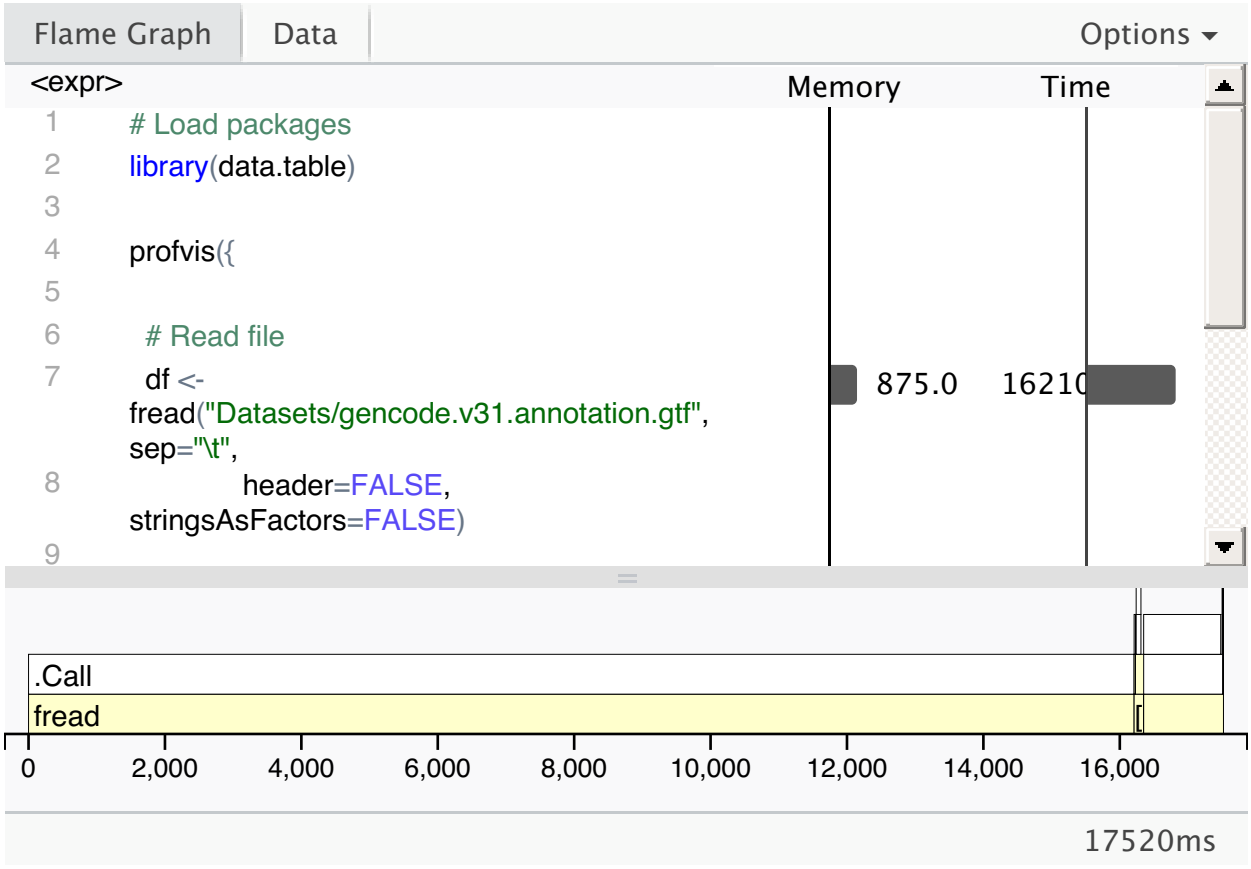
The following set of codes reads in the GTF file and then extracts the gene names for each gene.

```
# Load packages
library(profvis)

profvis({

  # Read file
  df <- read.table("Datasets/gencode.v31.annotation.gtf", sep="\t",
                   header=FALSE, stringsAsFactors=FALSE)

  # Subset required feature
  df <- df[which(df$V3=="gene"), ]

  # Retrieve gene names
  attr <- sapply(strsplit(df$V9, split=";"), function(x) {x[3]})

})
```




Clearly, reading in the file takes up majority of the time, i.e. this step is the bottleneck. This is because the file size is more than 1GB! As we have learned from *Efficient R Codes*, we can use `fread()` from **data.table** package to read in the file more efficiently.

```
# Load packages
library(data.table)

profvis({

  # Read file
  df <- fread("Datasets/gencode.v31.annotation.gtf", sep="\t",
              header=FALSE, stringsAsFactors=FALSE)

  # Subset required feature
  df <- df[which(df$V3=="gene"), ]

  # Retrieve gene names
  attr <- sapply(strsplit(df$V9, split=";"), function(x) {x[3]})

})
```




Replacing `read.table()` with `fread()` decreased time taken to read in the file, thus increasing overall efficiency.

## Reference

Gillespie, C. and Lovelace, R. 2017. *Efficient R Programming*. O'Reilly Media.