Dynamic Report - TFM

Marta Sanchez Delgado

5 de junio, 2018

Contents

1 Initial instructions **2**
1.1 Downloading expression file from GTEx . . . 2
1.2 General instructions to update input file . . . 2

2 Context **3**

3 Visualization of NUMTs in UCSC genome browser **3**

4 Installing only packages we need **4**

5 Input file format **5**

6 All intermediate files and the final table **6**
6.1 File 1 and 2: "All_attributes.txt" & "All_filters.txt" . . . 6
6.2 File 3: "gene_results.txt" . . . 7
6.3 File 4 and 5: "up_gene_results.txt" and "down_gene_results.txt" . . . 7
6.4 File 6: "genes.txt" . . . 8
6.5 File 7: "go_results.txt" . . . 8
6.6 File 8: "phenotype_results.txt" . . . 9
6.7 File 9: "mean_tpm_GTEx.txt" . . . 10
6.8 File 10: "subset_expressed.txt" . . . 10
6.9 File 11: "FINAL_OUTPUT_TABLE.txt" . . . 17

7 References **20**

1 Initial instructions

1.1 Downloading expression file from GTEx

The first step to correctly generate all expression information from your initial coordinates is to download the **`GTEx_Analysis_2016-01-15_v7_RNASeQCv1.1.8_gene_median_tpm.gct`** file from: https://gtexportal.org/home/datasets. Gene expression on the GTEx Portal are shown in Transcripts per Million (TPM), and the samples come from the 1000 genomes project. The downloaded file contains median gene counts (in TPM) by tissue (53 in total).

1.2 General instructions to update input file

All R-scripts are in the first part of the document **`Scripts-TFM.R`**. However, the corresponding **`Dynamic report-TFM.Rmd`** can be used to generate the new set of output files automatically.

> **IMPORTANT:** *Fist read 'Dynamic report-TFM.pdf' for **'input file format'**. Then, you need to copy your input document in the folder containing 'Dynamic report-TFM.Rmd' and indicate in the next lines your input CVS file name.*

To change the **input file** you need to change the name of the .csv (NUMTs_coord.csv) document in the begining of this document:

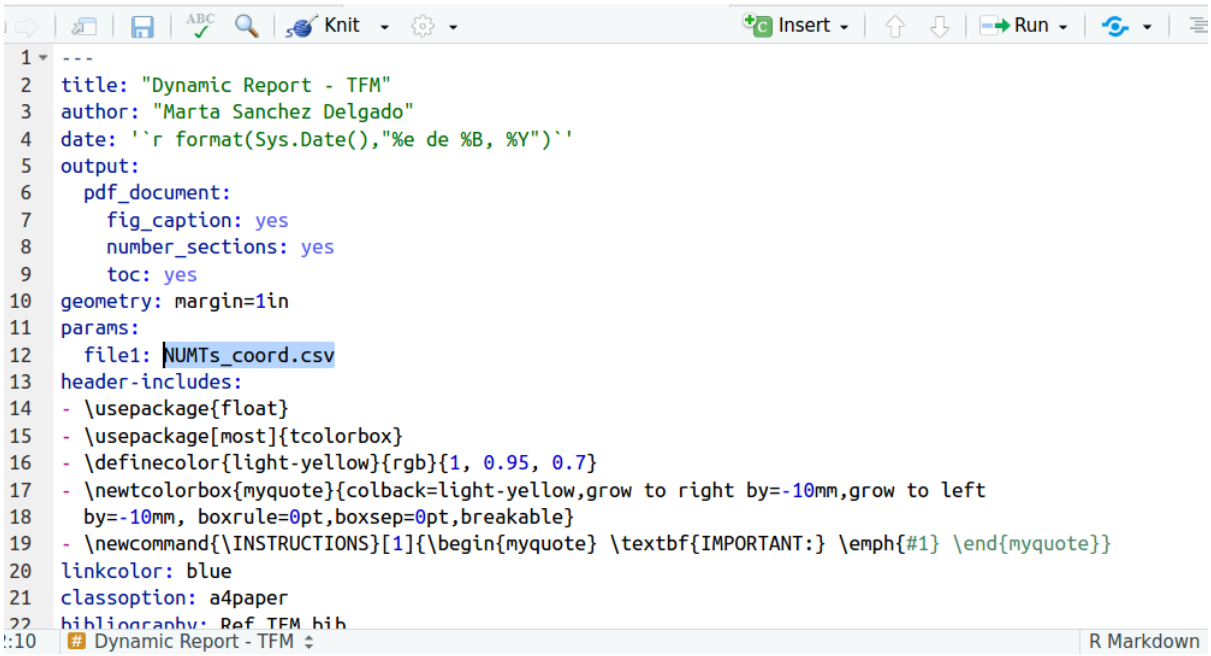

Figure 1: **Screenshot instruction to change the name of the 'input file'.** In the begining of the document you have the 'params' subsection, the 'file1:' corresponds to the 'input file' with the initial coordinates.

Now all the output files and all the information on `Dynamic report-TFM` documents will be generated with your new .csv data-table named NUMTs_coord.csv.

> **IMPORTANT:** *Once you have done this first step, you can generate your new .html, which will have all data updated by pressing 'Knit' (See next figure). Maybe this will take more than an hour.*

The following pages will be generated with the new information. Now you have all data, statistics and tables updated with the coordinates in document NUMTs_coord.csv.

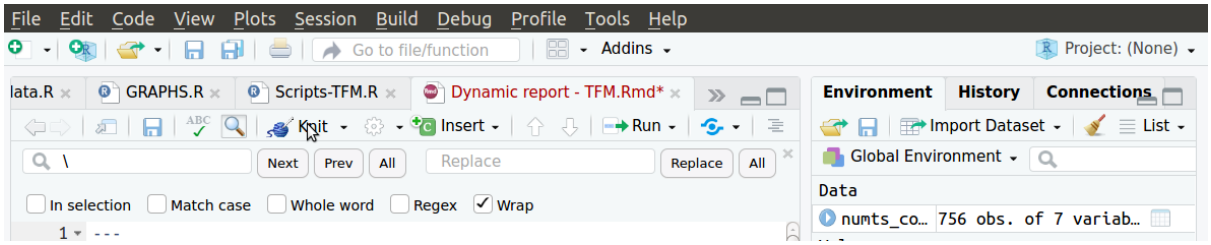

Figure 2: **Screenshot instruction for generate the new .html file.** In RStudio, by using R Markdown, we can generate the corresponding .html file by pression 'Knit' (in the upper-left part of the first square in RStudio).

2 Context

The following scripts were generated for the final master's project in Bioinformatics and Biostatistics (*Universitat Oberta de Catalunya*) entitle ***Are Nuclear Insertions of Mitochondrial Origin Pseudogenes?***.

This Master's project focussed on the study of Nuclear mitochondrial DNA sequences (NUMTs). NUMTs are the result of a continuous DNA transfer from mitochondria to the nucleus (Boogaart, Samallo, and Agsteribbe 1982; Tsuzuki et al. 1983).

In 2011, in a published work led by Dr Cristina Santos, it was identified 755 NUMTs in the human genome (Ramos et al. 2011). They compared the human mtDNA (NC_012920) against human genome (GRCh37/hg19 assembly), and they described different aspects of this comparisons: frequency, distribution and size of NUMTs for each chromosome; % identity between NUMTs and mtDNA sequence... Based on this information and **NUMTs coordinates**, in the present master's final project, we want to clarify whether or not these NUMTs origin pseudogenes.

The present dynamic report generates a set of intermediate (.txt documents) and a final file called **FINAL_OUTPUT_TABLE.txt** with all relevant genetic content and with a more in-depth expression and gene ontology study in the genes encoded in this NUMTs. Additionally, we also perform a small conservation study of these regions in the genome of other primates. By changing the **input document**, and following the next instruction, it is possible to generate a new set of intermediate and final documents with the new outputs.

3 Visualization of NUMTs in UCSC genome browser

First of all, with the NUMTs coordinates publically available by Ramos et al. (2011), we created the file `bed_NUMTs-ID.txt`, which can be uploaded to ***UCSC genome browser –> custom track*** to visualise our NUMTs.

The `bed_NUMTs-ID.txt` also include the corresponding mitochondrial regions for each NUMT, with the same NUMT ID by adding "mt" in the beginning.

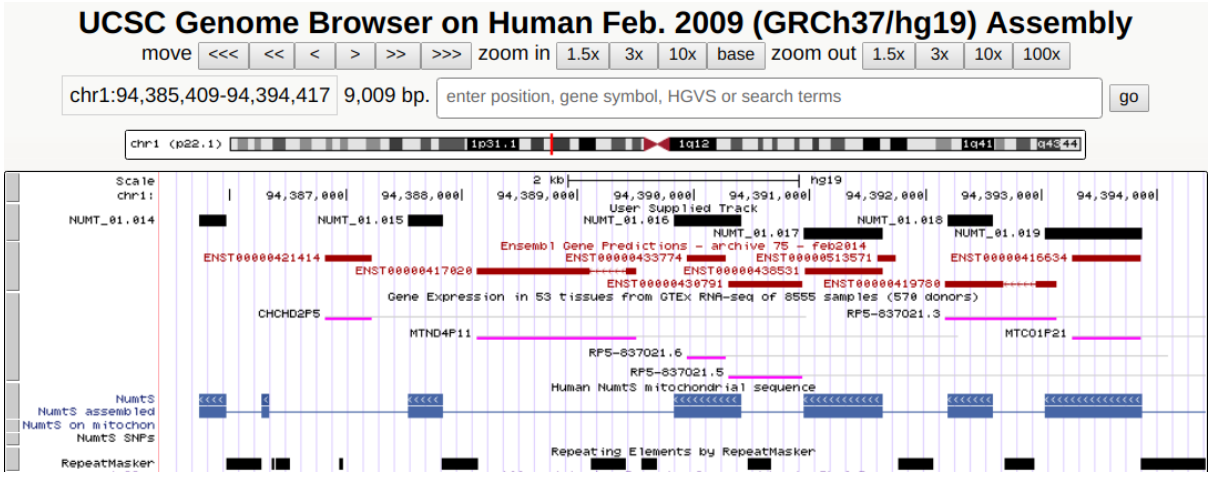

Figure 3: **Custom Track NUMNTs visualization in UCSC Genome Browser**. The first track on the picture is our Custom Track from the document "bed_NUMTsID.txt".

4 Installing only packages we need

Depending on the computer and session, we already have some R packages installed. To install only the ones we need we will use the following script:

```
######## PART 1: Scripts from Dynamic_report_TFM.Rmd ##########

# Installing package if needed ----
list.of.packages <- c("rstudioapi", "dplyr", "xlsx", "rJava", "gplots",
                      "devtools", "ggplot2")
new.packages <- list.of.packages[!(list.of.packages %in%
                                   installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)
```

However, to generate our data, we also need to download special packages from Bioconductor:

> **IMPORTANT:** *If you cannot install Bioconductor's* ***biomaRt*** *package by using* ***'biocLite("biomaRt")'****, in Linux, if you have administrative privileges, you can write in the command line:* ***'sudo apt-get install r-bioc-biomart'*** *to install it.*

5 Input file format

The document must be .csv with `comma (",")` separator, which is the one automatically used in most programmes like Excel or LibreOffice Calc when we save the data as .csv. The first line of the document will be the **Column names**. The First column will name as `id` (with any ID you wanted to use), second column `chr` (with the number of the corresponding chromosome), third `start_n` with the starting bp coordinate and then `end_n` with the end pb coordinate. The last three columns will be additional information. In the case of the original document created for this final master's project corresponds to the coordinates mapping these regions in the mitochondrial: 6th column entitle `mt` and in all cases "mt" because is how mitochondrial DNA is recognised, and then both, initial `start_mt` and final `end_mt` coordinates in the mitochondrial DNA.

```
## 'data.frame':    756 obs. of  7 variables:
##  $ id      : Factor w/ 756 levels "NUMT_01.001",..: 1 2 ...
##  $ chr     : Factor w/ 24 levels "1","10","11",..: 1 1 ...
##  $ start_n : int  564461 5614806 ...
##  $ end_n   : int  570304 5614937 ...
##  $ mt      : Factor w/ 1 level "mt": 1 1 ...
##  $ start_mt: int  3911 9453 ...
##  $ end_mt  : int  9755 9583 ...
```

```
##            id chr  start_n    end_n mt start_mt end_mt
## 1 NUMT_01.001   1   564461   570304 mt     3911   9755
## 2 NUMT_01.002   1  5614806  5614937 mt     9453   9583
## 3 NUMT_01.003   1  5910318  5910528 mt     2466   2675
## 4 NUMT_01.004   1  8969802  8969967 mt     8040   8205
## 5 NUMT_01.005   1  9634687  9634887 mt      907   1117
## 6 NUMT_01.006   1 11202904 11202975 mt    12293  12358
```

The file NUMTs_coord.csv contains a total of 756 rows with coordinates.

6 All intermediate files and the final table

6.1 File 1 and 2: "All_attributes.txt" & "All_filters.txt"

To download the list of genes within our initial coordinates and its associated phenotype description, gene ontology (GO) and conservation in other species, we use `biomaRt` package from `Bioconductor`:

6.1.1 The `biomaRt` package

```
##
## To cite the biomaRt package in publications use:
##
##   Mapping identifiers for the integration of genomic datasets with
##   the R/Bioconductor package biomaRt. Steffen Durinck, Paul T.
##   Spellman, Ewan Birney and Wolfgang Huber, Nature Protocols 4,
##   1184-1191 (2009).
##
##   BioMart and Bioconductor: a powerful link between biological
##   databases and microarray data analysis. Steffen Durinck, Yves
##   Moreau, Arek Kasprzyk, Sean Davis, Bart De Moor, Alvis Brazma
##   and Wolfgang Huber, Bioinformatics 21, 3439-3440 (2005).
##
## To see these entries in BibTeX format, use 'print(<citation>,
## bibtex=TRUE)', 'toBibtex(.)', or set
## 'options(citation.bibtex.max=999)'.
```

6.1.2 Preparing Package 'biomaRt'

We set up the dataset we will use, specifically, `ensembl_MART_ensembl`, which is working with the version:

It is automatically generated the first two files: `All_attributes.txt` (with all attributes you can download with `biomaRt` package) and `All_filters.txt` (with all `filters` to select the information to use from your input file).

The dimensions of File 1 are: 1416, 3 and the dimensions of File 2: 303, 2

6.2 File 3: "gene_results.txt"

Filtering by our initial coordinates, we create the file `gene_results.txt` with all genes which coordinates and our initial coordinates overlaps partially or totally.

```
##            id       chromosome_name start_position
##  NUMT_04.035:  11   2      :153     Min.   :   536816
##  NUMT_05.022:  11   7      : 70     1st Qu.: 38039816
##  NUMT_02.043:  10   X      : 63     Median : 80736542
##  NUMT_01.001:   8   1      : 59     Mean   : 86166624
##  NUMT_02.058:   8   4      : 58     3rd Qu.:120972370
##  NUMT_05.030:   8   (Other):511     Max.   :240713167
##  (Other)    :1099   NA's   :241     NA's   :241
##   end_position           strand            hgnc_symbol
##  Min.   :   564813   Min.   :-1.00000   MLPH     : 17
##  1st Qu.: 38078249   1st Qu.:-1.00000   LINC00630: 13
##  Median : 80925878   Median : 1.00000   LINC00882:  7
##  Mean   : 86268873   Mean   : 0.03939   ZNF540   :  7
##  3rd Qu.:120974671   3rd Qu.: 1.00000   ZNF571   :  7
##  Max.   :240775449   Max.   : 1.00000   (Other)  :480
##  NA's   :241         NA's   :241        NA's     :624
##        ensembl_gene_id_version        ensembl_gene_id transcript_count
##  ENSG00000115648.9 : 17        ENSG00000115648: 17    Min.   : 1.000
##  ENSG00000223546.2 : 13        ENSG00000223546: 13    1st Qu.: 1.000
##  ENSG00000171817.12:  7        ENSG00000171817:  7    Median : 1.000
##  ENSG00000180479.9 :  7        ENSG00000180479:  7    Mean   : 3.953
##  ENSG00000242759.2 :  7        ENSG00000242759:  7    3rd Qu.: 5.000
##  (Other)           :863        (Other)        :863    Max.   :32.000
##  NA's              :241        NA's           :241    NA's   :241
```

The dimensions of File 3 are: 1155, 9

In total, `gene_results.txt` contains 733 genes and 241/756 NUMTs do not overlap with any gene.

6.3 File 4 and 5: "up_gene_results.txt" and "down_gene_results.txt"

Genes in `gene_results.txt` also includes large gene coding proteins where the NUMTs are probably located in intronic regions. To eliminate this genes, an additional two other lists were generated with new coordinates obtained from the upstream or downstream part of the original NUMTs coordinates (between 100 - 1000 bp from the initial coordinates). Once we get this two new list of genes associated to different NUMTs, we eliminate from the initial list in `gene_results.txt` that genes also present upstream AND downstream the initial coordinates. However, we ALWAYS associated gene with NUMTs, and also those genes associated to specific NUMT is eliminate (to conserve genes which includes more than one NUMT).

```
## 'data.frame':    891 obs. of  9 variables:
##  $ id                     : Factor w/ 756 levels "NUMT_01.001",..: 1 1 ...
##  $ chromosome_name        : Factor w/ 24 levels "1","10","11",..: 1 1 ...
##  $ start_position         : int  536816 562757 ...
##  $ end_position           : int  659930 564390 ...
##  $ strand                 : int  -1 -1 ...
##  $ hgnc_symbol            : Factor w/ 275 levels "ABCA8","ACSM3",..: NA NA ...
##  $ ensembl_gene_id_version: Factor w/ 414 levels "ENSG00000003400.10",..: 262 213 ...
##  $ ensembl_gene_id        : Factor w/ 414 levels "ENSG00000003400",..: 262 213 ...
##  $ transcript_count       : int  5 1 ...
```

```
## 'data.frame':    905 obs. of  9 variables:
##  $ id                     : Factor w/ 756 levels "NUMT_01.001",..: 1 2 ...
##  $ chromosome_name        : Factor w/ 24 levels "1","10","11",..: 1 NA ...
##  $ start_position         : int  536816 NA ...
##  $ end_position           : int  659930 NA ...
```

```
##  $ strand                 : int  -1 NA ...
##  $ hgnc_symbol            : Factor w/ 274 levels "ABCA8","ACSM3",..: NA NA ...
##  $ ensembl_gene_id_version: Factor w/ 425 levels "ENSG00000003400.10",..: 264 NA ...
##  $ ensembl_gene_id        : Factor w/ 425 levels "ENSG00000003400",..: 264 NA ...
##  $ transcript_count       : int  5 NA ...
```

In the upstream part of NUMTs coordinates we find 414 and in the downstream region 425 genes. Once we have the three list of genes, we wanted to compare all of them and select the ones originated by a NUMT insertion.

The dimensions of File 4 are: 891, 9 and the dimensions of File 5: 905, 9

6.4 File 6: "genes.txt"

The distribution of the genes within NUMTs in the different chromosomes is:

```
##
##  1 10 11 12 13 14 15 16 17 18 19  2 20 21 22  3  4  5  6  7  8  9  X  Y
## 45 16 14  4  8  5  9 13 18  2  3 80  5  4  8 24 36 32 16 39 19 23 28  5
```

Of the 733 initial genes, after our filtering, we have 456 genes within the initial input coordinates.

The dimensions of File 6 are: 456, 1

6.5 File 7: "go_results.txt"

```
##    hgnc_symbol ensembl_gene_id_version      go_id            name_1006
## 1     MTRNR2L4       ENSG00000232196.2 GO:0005576 extracellular region
## 2     MTRNR2L4       ENSG00000232196.2 GO:0005737            cytoplasm
## 3     MTRNR2L5       ENSG00000249860.2 GO:0005576 extracellular region
## 4     MTRNR2L5       ENSG00000249860.2 GO:0005737            cytoplasm
## 5     MTRNR2L9       ENSG00000255633.3 GO:0005576 extracellular region
## 6     MTRNR2L9       ENSG00000255633.3 GO:0005737            cytoplasm
## 7     MTRNR2L8       ENSG00000255823.1 GO:0005576 extracellular region
## 8     MTRNR2L8       ENSG00000255823.1 GO:0005737            cytoplasm
## 9    MTRNR2L10       ENSG00000256045.1 GO:0005576 extracellular region
## 10   MTRNR2L10       ENSG00000256045.1 GO:0005737            cytoplasm
## 11    MTRNR2L3       ENSG00000256222.1 GO:0005576 extracellular region
## 12    MTRNR2L3       ENSG00000256222.1 GO:0005737            cytoplasm
## 13    MTRNR2L1       ENSG00000256618.1 GO:0005576 extracellular region
## 14    MTRNR2L1       ENSG00000256618.1 GO:0005737            cytoplasm
## 15    MTRNR2L7       ENSG00000256892.1 GO:0005576 extracellular region
## 16    MTRNR2L7       ENSG00000256892.1 GO:0005737            cytoplasm
## 17   MTRNR2L12       ENSG00000269028.2 GO:0005576 extracellular region
## 18   MTRNR2L12       ENSG00000269028.2 GO:0005737            cytoplasm
## 19    MTRNR2L6       ENSG00000270672.1 GO:0005576 extracellular region
## 20    MTRNR2L6       ENSG00000270672.1 GO:0005737            cytoplasm
## 21    MTRNR2L2       ENSG00000271043.1 GO:0005576 extracellular region
## 22    MTRNR2L2       ENSG00000271043.1 GO:0005737            cytoplasm
```

Only 11 of the 456 genes originated by NUMTs insertion are anotated in Gene Ontology Consortium. However, as we can see in Figure 3, they are associated with different GO terms.

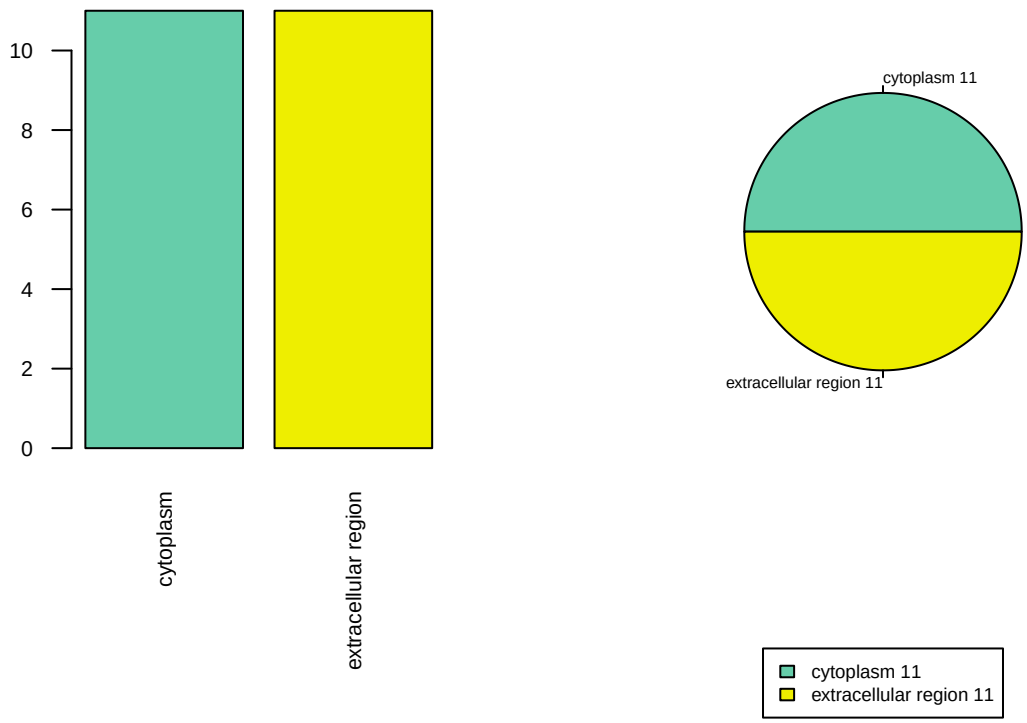

Figure 4: GO terms annotated for our list of genes.

The dimensions of File 7 are: 22, 5

6.6 File 8: "phenotype_results.txt"

```
##    hgnc_symbol       ensembl_gene_id_version transcript_count
##  MIR4461 :  2   ENSG00000198744.5:  1        Min.   :1.000
##  MIR4484 :  1   ENSG00000198868.3:  1        1st Qu.:1.000
##  MTATP6P1:  1   ENSG00000216713.1:  1        Median :1.000
##  MTATP6P2:  1   ENSG00000216853.1:  1        Mean   :1.002
##  MTATP6P3:  1   ENSG00000217044.1:  1        3rd Qu.:1.000
##  (Other) :182   ENSG00000217083.1:  1        Max.   :2.000
##  NA's    :268   (Other)          :450
##          gene_biotype
##  antisense     :  1
##  lincRNA       :  1
##  miRNA         :  4
##  protein_coding: 13
##  pseudogene    :436
##  snRNA         :  1
##
##                                                                          description
##  microRNA 4461 [Source:HGNC Symbol;Acc:41656]                                  :  2
##  hsa-mir-6723 [Source:miRBase;Acc:MI0022558]                                   :  1
##  microRNA 4484 [Source:HGNC Symbol;Acc:41799]                                  :  1
##  mitochondrially encoded ATP synthase 6 pseudogene 1 [Source:HGNC Symbol;Acc:44575]:  1
##  mitochondrially encoded ATP synthase 6 pseudogene 2 [Source:HGNC Symbol;Acc:44576]:  1
##  (Other)                                                                       :183
##  NA's                                                                          :267
```

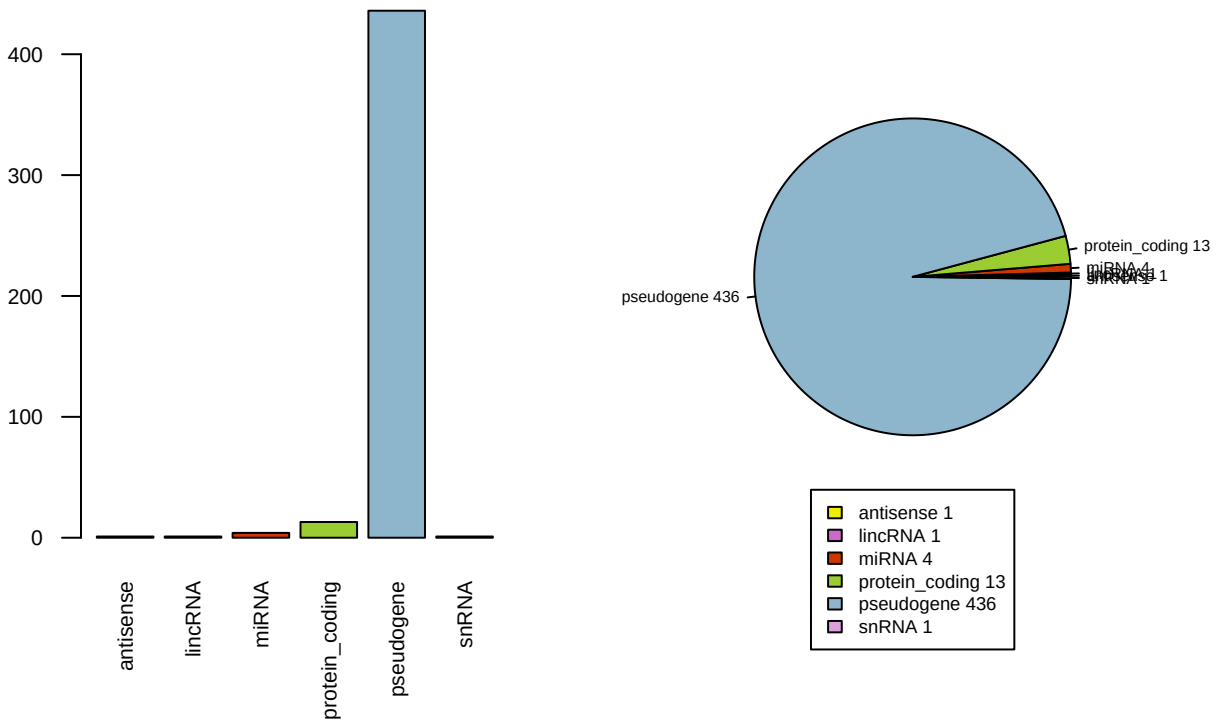

Figure 5: Gene biotype annotated for our list of genes.

For the 456 whithin our NUMTs, 456

In this case, all genes within our NUMTs are classified in ensembl-Biotype

The dimensions of File 8 are: 456, 5

6.7 File 9: "mean_tpm_GTEx.txt"

We then search in `GTEx_Analysis_2016-01-15_v7_RNASeQCv1.1.8_gene_median_tpm.gct` document our list of genes included in `genes.txt`.

Initially, we save all data existing for all our genes and calculate the mean and total TGM per gene.

The dimensions of File 9 are: 452, 58

6.8 File 10: "subset_expressed.txt"

In `nrow(mean_tpm_GTEx).txt` document, for the 456 within our initial coordinates, 452 have expression data in GTEx Portal. However, some of them are not expressed in any tissue. To subset the expressed genes, we will create an additional table containging genes with > 0.5 TMP in at least, one tissue:

```
##  Adipose...Subcutaneous Adipose...Visceral..Omentum. Adrenal.Gland
##  Min.   :   0.0000      Min.   :   0.000             Min.   :   0.000
##  1st Qu.:   0.0770      1st Qu.:   0.025             1st Qu.:   0.000
##  Median :   0.1716      Median :   0.149             Median :   0.151
##  Mean   :  58.5435      Mean   :  69.756             Mean   :  92.348
##  3rd Qu.:   0.9008      3rd Qu.:   0.930             3rd Qu.:   1.105
##  Max.   :3108.0000      Max.   :3881.000             Max.   :5478.500
##  Artery...Aorta      Artery...Coronary   Artery...Tibial
##  Min.   :   0.0000   Min.   :   0.0000   Min.   :   0.0000
##  1st Qu.:   0.0847   1st Qu.:   0.0338   1st Qu.:   0.0000
##  Median :   0.1850   Median :   0.1776   Median :   0.1688
##  Mean   :  30.3727   Mean   :  38.8836   Mean   :  33.3114
```

```
##  3rd Qu.:   0.6637   3rd Qu.:   0.7224   3rd Qu.:   0.7664
##  Max.   :1767.0000   Max.   :2189.0000   Max.   :1917.0000
##     Bladder          Brain...Amygdala
##  Min.   :   0.0000   Min.   :   0.000
##  1st Qu.:   0.0000   1st Qu.:   0.000
##  Median :   0.2607   Median :   0.096
##  Mean   :  47.6647   Mean   : 108.916
##  3rd Qu.:   1.0280   3rd Qu.:   1.393
##  Max.   :2810.0000   Max.   :6332.500
##  Brain...Anterior.cingulate.cortex..BA24. Brain...Caudate..basal.ganglia.
##  Min.   :   0.000                          Min.   :   0.000
##  1st Qu.:   0.000                          1st Qu.:   0.000
##  Median :   0.090                          Median :   0.128
##  Mean   : 107.712                          Mean   : 128.959
##  3rd Qu.:   1.179                          3rd Qu.:   1.656
##  Max.   :6276.000                          Max.   :7405.000
##  Brain...Cerebellar.Hemisphere Brain...Cerebellum Brain...Cortex
##  Min.   :   0.000              Min.   :   0.000   Min.   :   0.000
##  1st Qu.:   0.000              1st Qu.:   0.072   1st Qu.:   0.000
##  Median :   0.513              Median :   0.535   Median :   0.106
##  Mean   :  66.819              Mean   :  78.096   Mean   : 106.524
##  3rd Qu.:   1.500              3rd Qu.:   1.649   3rd Qu.:   1.288
##  Max.   :4014.500              Max.   :4581.000   Max.   :6114.500
##  Brain...Frontal.Cortex..BA9. Brain...Hippocampus Brain...Hypothalamus
##  Min.   :   0.000             Min.   :   0.000    Min.   :   0.000
##  1st Qu.:   0.000             1st Qu.:   0.000    1st Qu.:   0.000
##  Median :   0.097             Median :   0.095    Median :   0.127
##  Mean   :  91.629             Mean   : 114.700    Mean   : 105.871
##  3rd Qu.:   1.076             3rd Qu.:   1.387    3rd Qu.:   1.257
##  Max.   :5363.000             Max.   :6642.000    Max.   :6173.000
##  Brain...Nucleus.accumbens..basal.ganglia. Brain...Putamen..basal.ganglia.
##  Min.   :   0.000                           Min.   :   0.000
##  1st Qu.:   0.000                           1st Qu.:   0.000
##  Median :   0.156                           Median :   0.083
##  Mean   : 114.504                           Mean   : 134.231
##  3rd Qu.:   1.483                           3rd Qu.:   1.594
##  Max.   :6691.000                           Max.   :7739.000
##  Brain...Spinal.cord..cervical.c.1. Brain...Substantia.nigra
##  Min.   :   0.000                   Min.   :   0.000
##  1st Qu.:   0.000                   1st Qu.:   0.000
##  Median :   0.180                   Median :   0.082
##  Mean   :  82.647                   Mean   : 109.764
##  3rd Qu.:   1.136                   3rd Qu.:   1.249
##  Max.   :4649.000                   Max.   :6209.000
##  Breast...Mammary.Tissue Cells...EBV.transformed.lymphocytes
##  Min.   :   0.000        Min.   :   0.0000
##  1st Qu.:   0.075        1st Qu.:   0.0689
##  Median :   0.178        Median :   0.1672
##  Mean   :  62.097        Mean   :  27.4373
##  3rd Qu.:   0.931        3rd Qu.:   0.6375
##  Max.   :3424.500        Max.   :1650.5000
##  Cells...Transformed.fibroblasts Cervix...Ectocervix Cervix...Endocervix
##  Min.   :   0.0000               Min.   :   0.0000   Min.   :   0.0000
##  1st Qu.:   0.0000               1st Qu.:   0.0712   1st Qu.:   0.0525
##  Median :   0.1602               Median :   0.2128   Median :   0.3654
##  Mean   :  23.3810               Mean   :  31.2416   Mean   :  36.8054
##  3rd Qu.:   0.6394               3rd Qu.:   0.5674   3rd Qu.:   0.8342
##  Max.   :1379.0000               Max.   :1818.5000   Max.   :2113.0000
```

```
##  Colon...Sigmoid    Colon...Transverse
##  Min.   :   0.000   Min.   :   0.000
##  1st Qu.:   0.000   1st Qu.:   0.027
##  Median :   0.241   Median :   0.198
##  Mean   :  62.345   Mean   :  73.790
##  3rd Qu.:   1.560   3rd Qu.:   1.222
##  Max.   :3499.000   Max.   :4288.500
##  Esophagus...Gastroesophageal.Junction Esophagus...Mucosa
##  Min.   :   0.000                      Min.   :   0.0000
##  1st Qu.:   0.000                      1st Qu.:   0.0000
##  Median :   0.221                      Median :   0.1515
##  Mean   :  62.036                      Mean   :  34.0182
##  3rd Qu.:   1.227                      3rd Qu.:   0.8776
##  Max.   :3503.000                      Max.   :1995.0000
##  Esophagus...Muscularis Fallopian.Tube      Heart...Atrial.Appendage
##  Min.   :   0.000       Min.   :   0.0000   Min.   :   0.000
##  1st Qu.:   0.000       1st Qu.:   0.0000   1st Qu.:   0.026
##  Median :   0.192       Median :   0.1970   Median :   0.159
##  Mean   :  63.550       Mean   :  39.7982   Mean   : 114.793
##  3rd Qu.:   1.092       3rd Qu.:   0.9203   3rd Qu.:   1.566
##  Max.   :3619.500       Max.   :2240.0000   Max.   :6837.000
##  Heart...Left.Ventricle Kidney...Cortex        Liver
##  Min.   :   0.000       Min.   :   0.000   Min.   :   0.000
##  1st Qu.:   0.000       1st Qu.:   0.000   1st Qu.:   0.000
##  Median :   0.106       Median :   0.111   Median :   0.101
##  Mean   : 138.808       Mean   : 105.737   Mean   :  89.514
##  3rd Qu.:   1.867       3rd Qu.:   1.099   3rd Qu.:   1.591
##  Max.   :8294.000       Max.   :6276.000   Max.   :5360.000
##       Lung           Minor.Salivary.Gland Muscle...Skeletal
##  Min.   :   0.0000   Min.   :   0.0000    Min.   :   0.000
##  1st Qu.:   0.1118   1st Qu.:   0.0516    1st Qu.:   0.000
##  Median :   0.2510   Median :   0.2162    Median :   0.072
##  Mean   :  38.2883   Mean   :  43.3947    Mean   :  81.281
##  3rd Qu.:   1.0390   3rd Qu.:   0.9248    3rd Qu.:   1.391
##  Max.   :2172.0000   Max.   :2473.0000    Max.   :4844.500
##  Nerve...Tibial          Ovary              Pancreas
##  Min.   :   0.0000   Min.   :   0.0000   Min.   :   0.0000
##  1st Qu.:   0.1156   1st Qu.:   0.0748   1st Qu.:   0.0000
##  Median :   0.3490   Median :   0.2573   Median :   0.0933
##  Mean   :  40.9083   Mean   :  37.8839   Mean   :  25.3360
##  3rd Qu.:   1.6265   3rd Qu.:   1.2463   3rd Qu.:   0.4781
##  Max.   :2221.0000   Max.   :2238.0000   Max.   :1411.0000
##    Pituitary            Prostate
##  Min.   :   0.0000   Min.   :   0.000
##  1st Qu.:   0.0000   1st Qu.:   0.000
##  Median :   0.2112   Median :   0.237
##  Mean   :  43.4484   Mean   :  69.723
##  3rd Qu.:   0.8588   3rd Qu.:   0.847
##  Max.   :2510.0000   Max.   :4062.500
##  Skin...Not.Sun.Exposed..Suprapubic. Skin...Sun.Exposed..Lower.leg.
##  Min.   :   0.0000                   Min.   :   0.0000
##  1st Qu.:   0.1222                   1st Qu.:   0.1173
##  Median :   0.2482                   Median :   0.2667
##  Mean   :  49.3496                   Mean   :  45.4228
##  3rd Qu.:   1.0835                   3rd Qu.:   1.0277
##  Max.   :2779.0000                   Max.   :2536.0000
##  Small.Intestine...Terminal.Ileum     Spleen             Stomach
##  Min.   :   0.000                 Min.   :   0.0000   Min.   :   0.000
```

```
##  1st Qu.:   0.080                 1st Qu.:   0.0743   1st Qu.:   0.000
##  Median :   0.206                 Median :   0.1682   Median :   0.178
##  Mean   :  70.835                 Mean   :  40.7322   Mean   :  74.691
##  3rd Qu.:   1.375                 3rd Qu.:   0.7227   3rd Qu.:   1.343
##  Max.   :4085.000                 Max.   :2440.0000   Max.   :4273.500
##      Testis             Thyroid              Uterus
##  Min.   :   0.0000   Min.   :   0.0000   Min.   :   0.0000
##  1st Qu.:   0.2186   1st Qu.:   0.0541   1st Qu.:   0.0653
##  Median :   0.5612   Median :   0.3035   Median :   0.3013
##  Mean   :  48.9623   Mean   :  52.4518   Mean   :  38.8625
##  3rd Qu.:   1.0517   3rd Qu.:   0.9350   3rd Qu.:   0.8382
##  Max.   :2798.0000   Max.   :3056.5000   Max.   :2252.0000
##      Vagina           Whole.Blood        tissue_means
##  Min.   :   0.0000   Min.   :  0.0000   Min.   :   0.012
##  1st Qu.:   0.1260   1st Qu.:  0.0000   1st Qu.:   0.088
##  Median :   0.2664   Median :  0.0769   Median :   0.240
##  Mean   :  34.3699   Mean   :  8.2614   Mean   :  66.619
##  3rd Qu.:   0.8346   3rd Qu.:  0.3096   3rd Qu.:   1.100
##  Max.   :1989.0000   Max.   :487.0000   Max.   :3854.066
##       sum
##  Min.   :     0.65
##  1st Qu.:     4.74
##  Median :    12.99
##  Mean   :  3597.43
##  3rd Qu.:    59.41
##  Max.   :208119.57
```

The dimensions of File 10 are: 72, 58

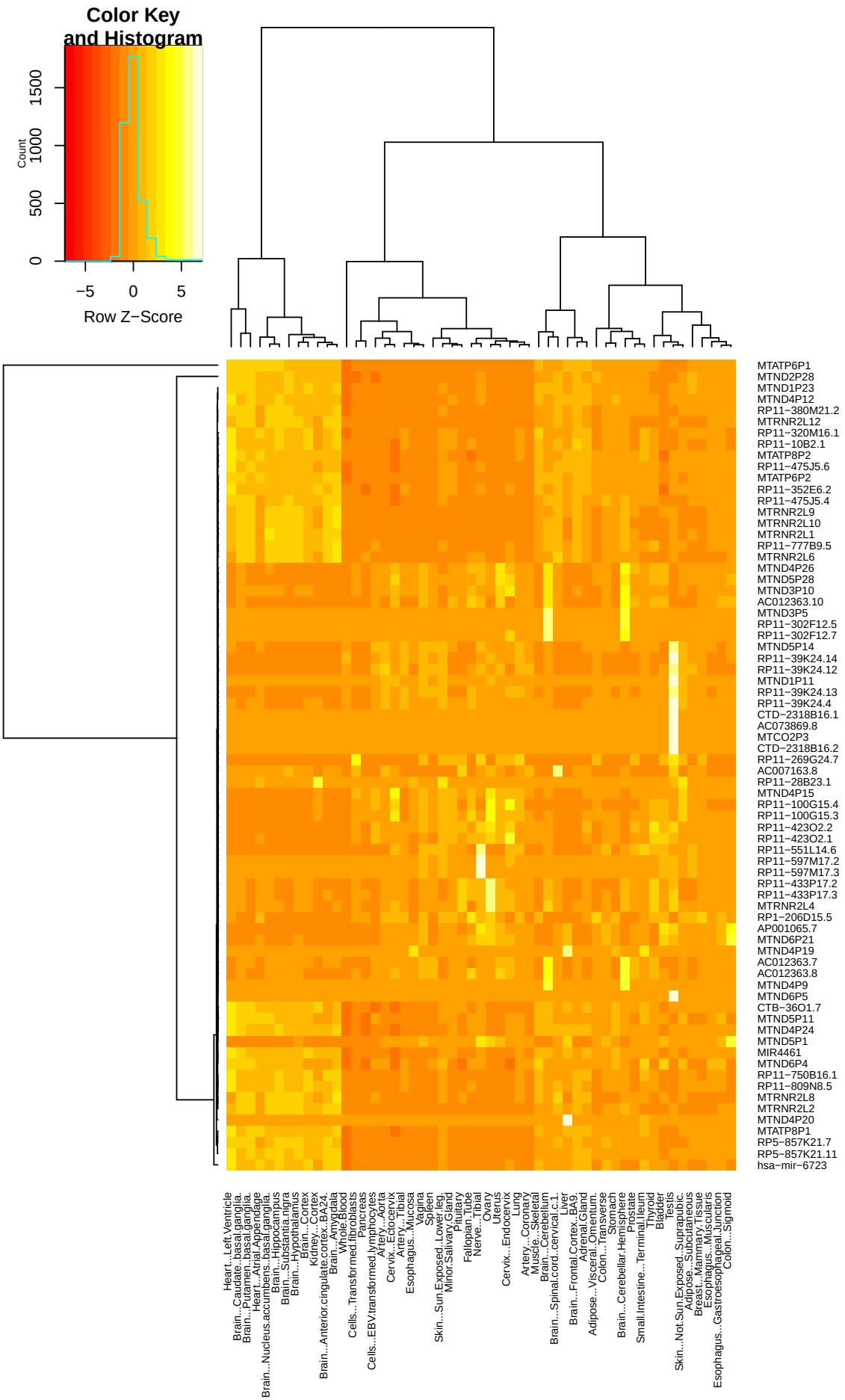

Figure 6: Heatmap of all expressed genes normalized by row (to see the different expression profile of each gene for the different tissues).

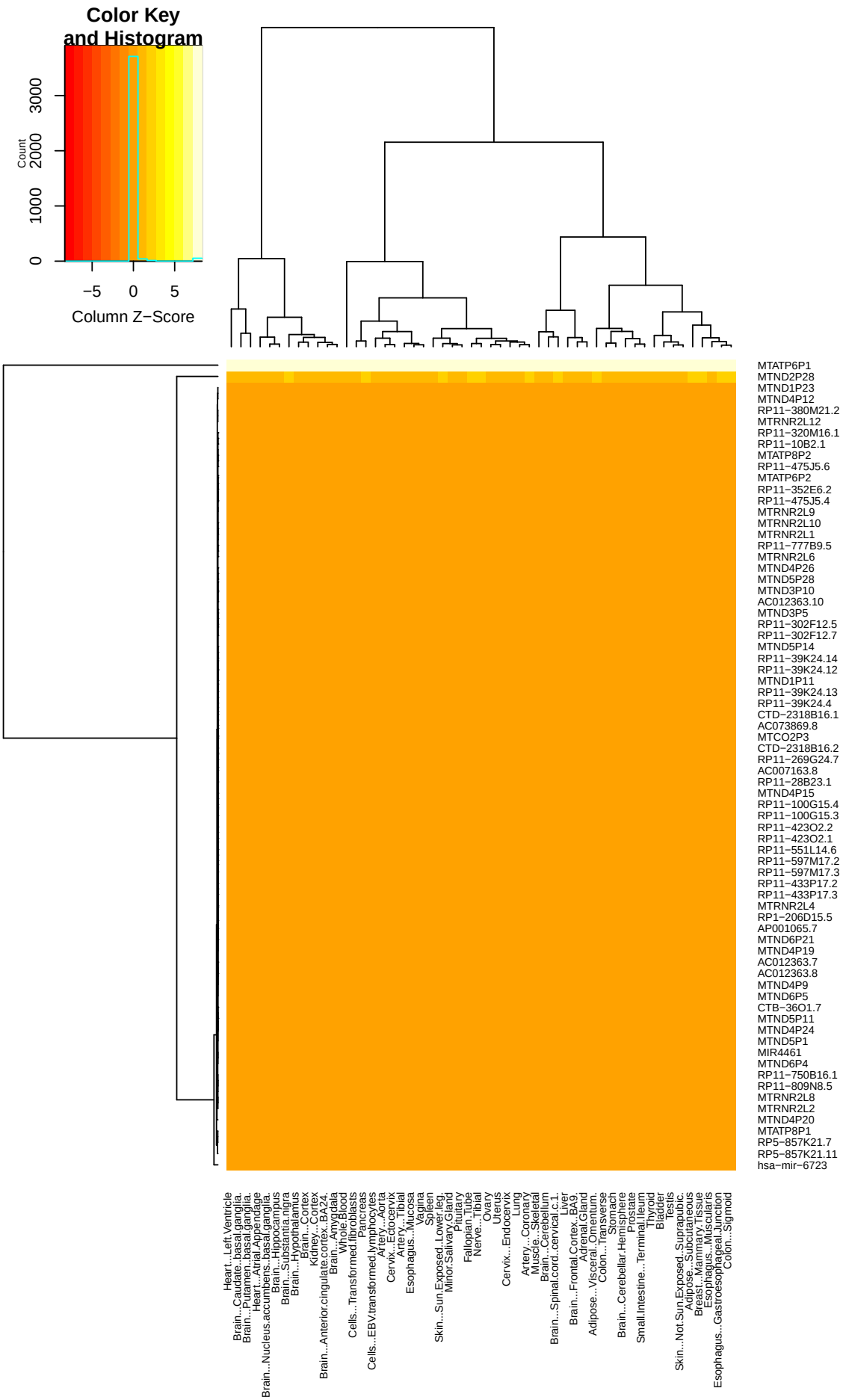

Figure 7: Heatmap of all expressed genes normalized by column (to see the different expression profile of the different gene for each tissue).

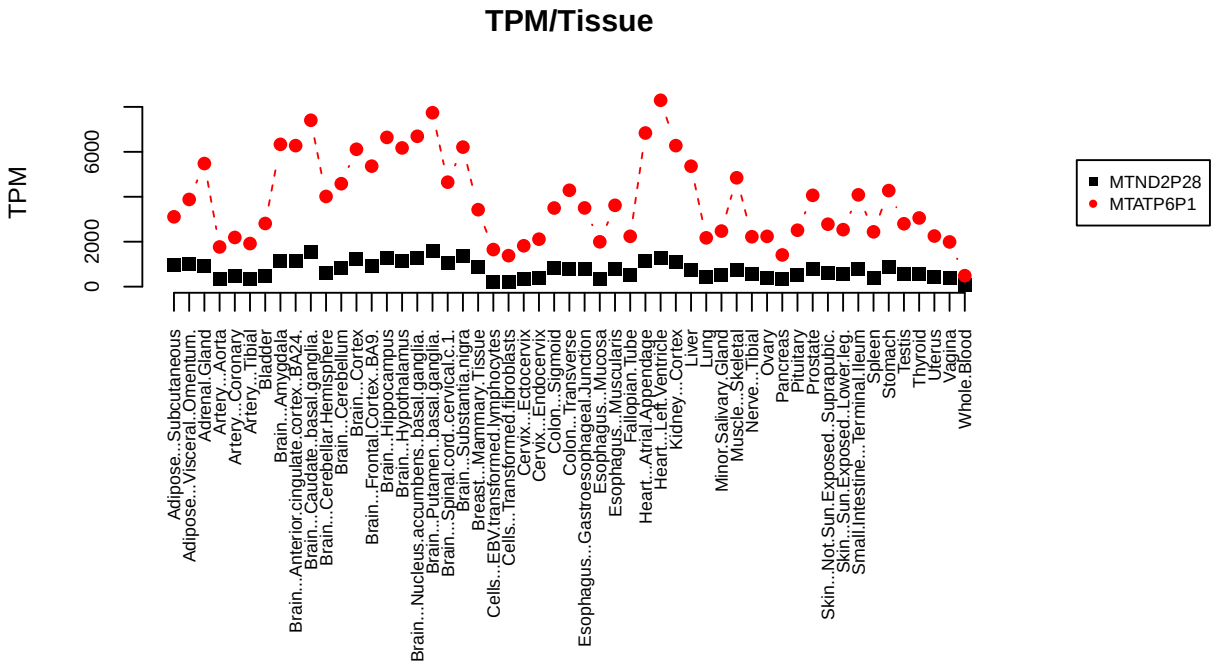

Figure 8: Graphical representation: expression profile of hight expressed genes (>=1000 TMP).

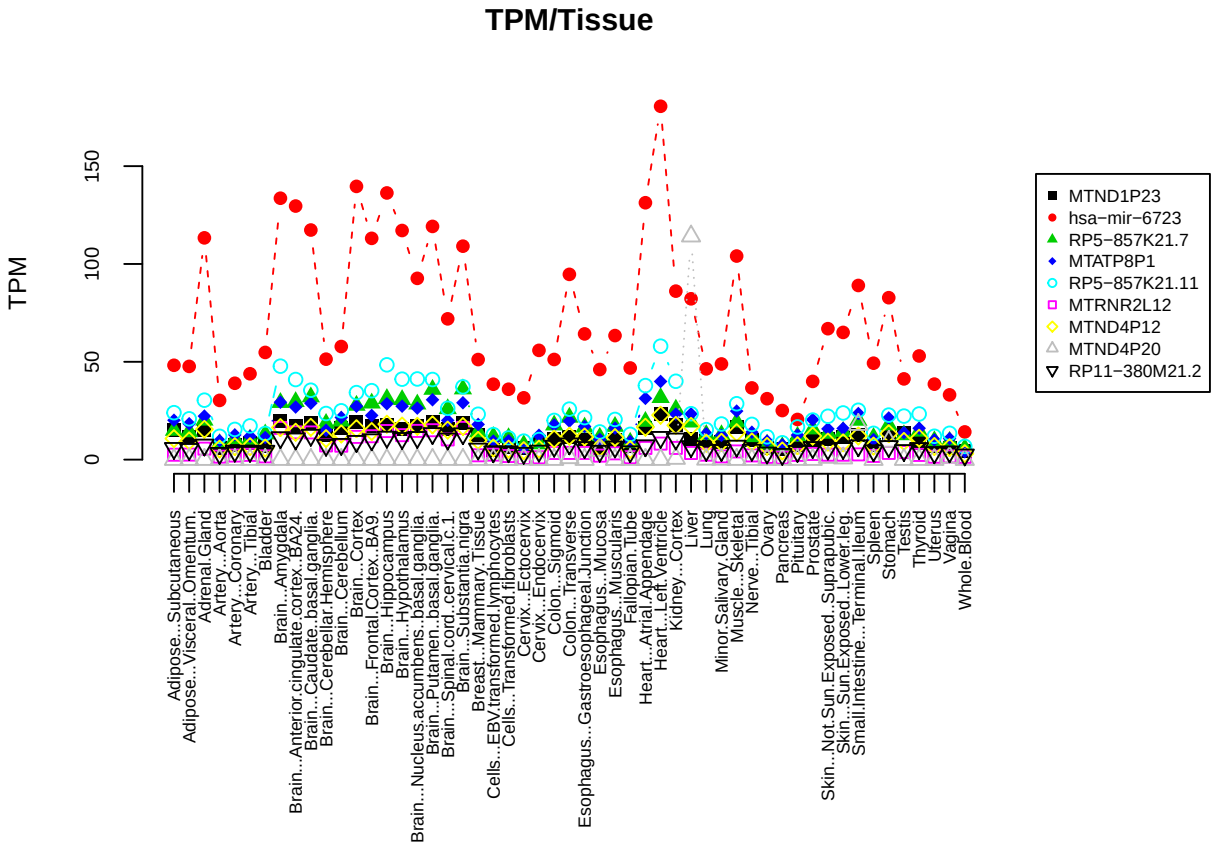

Figure 9: Graphical representation: expression profile of medium expressed genes (between 10 and 1000 TMP).

Of the 452 of our set of genes included in GTEx Portal, 72 are expressed in at least, one tissue (with >= 0.5 TPM). But the EMBL-EBI Expression Atlas classified genes in: - low expressed (between 0.5 and 10 TPM), - medium expressed (>=10 to 1000 TPM) and - high expressed (more than 1000 TPM).

In total, we have 2 high expressed (Figure) and 9 genes medium expressed (Figure)

6.9 File 11: "FINAL_OUTPUT_TABLE.txt"

Finally, the final table `FINAL_OUTPUT_TABLE.txt` will include relevant information from all the analysis perform in this Dynamic Report. The first 6 lines of our finall document will be:

The dimensions of File 11 are: 998, 75

```
##            id localization chr start_n  end_n mt start_mt end_mt
## 1 NUMT_01.001     intronic   1  564461 570304 mt     3911   9755
## 2 NUMT_01.001     intronic   1  564461 570304 mt     3911   9755
## 3 NUMT_01.001     intronic   1  564461 570304 mt     3911   9755
## 4 NUMT_01.001     intronic   1  564461 570304 mt     3911   9755
## 5 NUMT_01.001     intronic   1  564461 570304 mt     3911   9755
## 6 NUMT_01.001     intronic   1  564461 570304 mt     3911   9755
##   hgnc_symbol  Description gene_biotype name_1006 ensembl_gene_id_version
## 1    MTND1P23     MTND1P23   pseudogene      <NA>       ENSG00000225972.1
## 2    MTND2P28     MTND2P28   pseudogene      <NA>       ENSG00000225630.1
## 3        <NA> hsa-mir-6723   pseudogene      <NA>       ENSG00000237973.1
## 4        <NA> RP5-857K21.7   pseudogene      <NA>       ENSG00000229344.1
## 5    MTATP8P1     MTATP8P1   pseudogene      <NA>       ENSG00000240409.1
## 6    MTATP6P1     MTATP6P1   pseudogene      <NA>       ENSG00000248527.1
##   transcript_count tissue_means         sum GTEx_gene_id_version
## 1                1     11.81000    637.7400    ENSG00000225972.1
## 2                1    743.85943  40168.4094    ENSG00000225630.1
## 3                1     70.02962   3781.5996    ENSG00000237973.1
## 4                1     17.58864    949.7866    ENSG00000229344.1
## 5                1     18.09815    977.3002    ENSG00000240409.1
## 6                1   3854.06604 208119.5660    ENSG00000248527.1
##   Adipose...Subcutaneous Adipose...Visceral..Omentum. Adrenal.Gland
## 1                  14.90                        11.65        14.900
## 2                 941.20                       989.00       907.000
## 3                  48.22                        47.65       113.350
## 4                  14.90                        14.15        20.115
## 5                  20.14                        18.35        22.235
## 6                3108.00                      3881.00      5478.500
##   Artery...Aorta Artery...Coronary Artery...Tibial  Bladder
## 1          6.127              7.57           8.285    7.068
## 2        324.700            492.50         354.500  483.100
## 3         30.280             39.04          43.870   54.780
## 4          8.122             10.61          10.400   13.040
## 5          9.655             12.64          11.930   11.290
## 6       1767.000           2189.00        1917.000 2810.000
##   Brain...Amygdala Brain...Anterior.cingulate.cortex..BA24.
## 1           19.735                                   17.41
## 2         1152.500                                 1152.00
## 3          133.600                                  129.60
## 4           29.125                                   29.60
## 5           29.345                                   26.89
## 6         6332.500                                 6276.00
##   Brain...Caudate..basal.ganglia. Brain...Cerebellar.Hemisphere
## 1                          18.905                        12.685
## 2                        1555.500                       609.750
## 3                         117.350                        51.345
## 4                          31.985                        18.430
## 5                          28.835                        15.000
## 6                        7405.000                      4014.500
```

```
##   Brain...Cerebellum Brain...Cortex Brain...Frontal.Cortex..BA9.
## 1              16.01         19.405                       17.34
## 2             830.60       1226.500                      938.60
## 3              57.81        139.650                      113.10
## 4              19.61         27.690                       28.83
## 5              21.56         27.285                       22.59
## 6            4581.00       6114.500                     5363.00
##   Brain...Hippocampus Brain...Hypothalamus
## 1               17.79                15.52
## 2             1259.00              1132.00
## 3              136.30               117.10
## 4               31.15                30.63
## 5               28.54                27.31
## 6             6642.00              6173.00
##   Brain...Nucleus.accumbens..basal.ganglia.
## 1                                     16.96
## 2                                   1258.00
## 3                                     92.65
## 4                                     28.33
## 5                                     26.34
## 6                                   6691.00
##   Brain...Putamen..basal.ganglia. Brain...Spinal.cord..cervical.c.1.
## 1                          19.175                             15.74
## 2                        1584.500                           1065.00
## 3                         119.200                             71.95
## 4                          35.845                             25.97
## 5                          30.600                             20.05
## 6                        7739.000                           4649.00
##   Brain...Substantia.nigra Breast...Mammary.Tissue
## 1                   18.965                  12.810
## 2                 1369.000                 885.850
## 3                  109.100                  51.145
## 4                   36.065                  14.110
## 5                   29.060                  17.995
## 6                 6209.000                3424.500
##   Cells...EBV.transformed.lymphocytes Cells...Transformed.fibroblasts
## 1                              4.9045                           6.306
## 2                            214.1500                         201.900
## 3                             38.5500                          35.980
## 4                             11.7400                          11.150
## 5                              9.4795                           9.593
## 6                           1650.5000                        1379.000
##   Cervix...Ectocervix Cervix...Endocervix Colon...Sigmoid
## 1              5.8515               8.977           10.64
## 2            343.5500             406.100          820.00
## 3             31.5950              55.840           51.19
## 4              7.5350               7.978           17.63
## 5              6.1220              12.430           17.71
## 6           1818.5000            2113.000         3499.00
##   Colon...Transverse Esophagus...Gastroesophageal.Junction
## 1             11.685                                10.059
## 2            796.650                               783.550
## 3             94.690                                64.255
## 4             21.585                                16.555
## 5             19.725                                16.950
## 6           4288.500                              3503.000
##   Esophagus...Mucosa Esophagus...Muscularis Fallopian.Tube
## 1              5.892                 9.8765          6.905
```

```
## 2            330.600               781.4500        507.000
## 3             46.060                63.4000         46.840
## 4             11.350                16.9150          8.796
## 5             10.320                16.1100          9.232
## 6           1995.000              3619.5000       2240.000
##   Heart...Atrial.Appendage Heart...Left.Ventricle Kidney...Cortex   Liver
## 1                    16.28                  23.28           17.89   10.46
## 2                  1123.00                1274.00         1076.00  754.30
## 3                   131.30                 180.60           86.12   82.20
## 4                    19.48                  31.66           25.74   19.13
## 5                    31.32                  39.95           23.26   23.38
## 6                  6837.00                8294.00         6276.00 5360.00
##       Lung Minor.Salivary.Gland Muscle...Skeletal Nerve...Tibial    Ovary
## 1    9.282                8.829            16.825        10.0085    5.858
## 2  444.700              512.100           752.100       582.7500  382.900
## 3   46.400               48.900           104.050        36.6450   31.140
## 4   12.760               12.650            19.170        11.6500    7.806
## 5   13.220               12.410            24.810        14.0150    9.465
## 6 2172.000             2473.000          4844.500      2221.0000 2238.000
##    Pancreas Pituitary Prostate Skin...Not.Sun.Exposed..Suprapubic.
## 1    5.5455     7.311    9.680                                10.33
## 2  340.5500   514.000  805.200                               597.10
## 3   25.0950    20.530   39.980                                66.93
## 4    6.5620    11.850   15.595                                13.39
## 5    7.5835    12.560   20.450                                15.73
## 6 1411.0000  2510.000 4062.500                              2779.00
##   Skin...Sun.Exposed..Lower.leg. Small.Intestine...Terminal.Ileum   Spleen
## 1                          11.57                           12.79    7.647
## 2                         555.30                          772.20  366.550
## 3                          65.01                           89.01   49.295
## 4                          14.31                           20.03   11.250
## 5                          15.96                           23.55   10.225
## 6                        2536.00                         4085.00 2440.000
##    Stomach  Testis   Thyroid   Uterus   Vagina Whole.Blood
## 1   12.655   13.64    8.7245    6.845    6.069       4.364
## 2  884.200  566.70  557.8000  433.800  377.200      56.350
## 3   82.760   41.27   52.9750   38.600   33.100      14.170
## 4   17.605   12.63   14.4550    9.581    9.288       5.665
## 5   21.275   13.79   16.0350   10.120   10.740       4.042
## 6 4273.500 2798.00 3056.5000 2252.000 1989.000     487.000
##                                                                    description
## 1                                MT-ND1 pseudogene 23 [Source:HGNC Symbol;Acc:42092]
## 2                                MT-ND2 pseudogene 28 [Source:HGNC Symbol;Acc:42129]
## 3                                        hsa-mir-6723 [Source:miRBase;Acc:MI0022558]
## 4                                                                              <NA>
## 5 mitochondrially encoded ATP synthase 8 pseudogene 1 [Source:HGNC Symbol;Acc:44571]
## 6 mitochondrially encoded ATP synthase 6 pseudogene 1 [Source:HGNC Symbol;Acc:44575]
##   chromosome_name start_position end_position strand
## 1               1         564442       564813      1
## 2               1         565020       566063      1
## 3               1         566454       567996      1
## 4               1         568137       568818      1
## 5               1         568915       569121      1
## 6               1         569076       569756      1
```

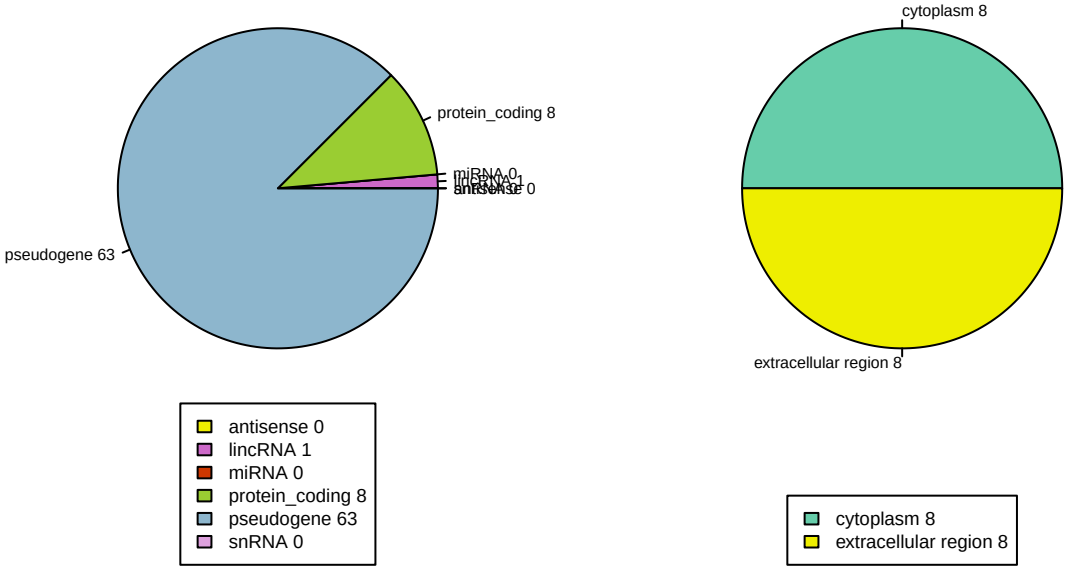

Figure 10: Gene biotype and GO term annotated for our list of expressed genes.

7 References

Boogaart, Paul van den, John Samallo, and Etienne Agsteribbe. 1982. "Similar Genes for a Mitochondrial Atpase Subunit in the Nuclear and Mitochondrial Genomes of Neurospora Crassa." *Nature* 298 (5870). Nature Publishing Group: 187.

Ramos, Amanda, Elena Barbena, Ligia Mateiu, María del Mar González, Quim Mairal, Manuela Lima, Rafael Montiel, Maria Pilar Aluja, and Cristina Santos. 2011. "Nuclear Insertions of Mitochondrial Origin: Database Updating and Usefulness in Cancer Studies." *Mitochondrion* 11 (6). Elsevier: 946–53.

Tsuzuki, Teruhisa, Hisayuki Nomiyama, Chiaki Setoyama, Shuichiro Maeda, and Kazunori Shimada. 1983. "Presence of Mitochondrial-Dna-Like Sequences in the Human Nuclear Dna." *Gene* 25 (2). Elsevier: 223–29.

```r
# General knitr options for RMarkdown ----
knitr::opts_chunk$set(external=TRUE, warning=FALSE, message=FALSE,
                      fig.align='center', fig.pos='H')
# Setting working directory ----
## IN R-STUDIO:
### Session --> Set working directory --> Choose directory
## WORKING DIRECTORY (THIS FOLDER):
library(rstudioapi)
current_path <- getActiveDocumentContext()$path
setwd(dirname(current_path))
getwd() # to show the pathway

######## PART 1: Scripts from Dynamic_report_TFM.Rmd ##########

# Installing package if needed ----
list.of.packages <- c("rstudioapi", "dplyr", "xlsx", "rJava", "gplots",
                      "devtools", "ggplot2")
new.packages <- list.of.packages[!(list.of.packages %in%
                                   installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)
# Conneting with Bioconductor ----
source("https://bioconductor.org/biocLite.R")
# Installing & loading Bioconductor packages ----
biocLite()
biocLite("biomaRt")
## If biocLite("biomaRt") do not work:
### LINUX COMAND LINE: sudo apt-get install r-bioc-biomart
# Uploading .csv input file ----
numts_coord <- read.csv(file=params$file1, sep = ",", header = TRUE)
str(numts_coord, vec.len = 1)
head(numts_coord)
# The biomaRt package ----
library("biomaRt")
citation("biomaRt") # Package citation
## Preparing Package 'biomaRt'
gene_mart = useMart(biomart="ENSEMBL_MART_ENSEMBL",
                    host="grch37.ensembl.org",
                    path="/biomart/martservice",
                    dataset="hsapiens_gene_ensembl")

listMarts(gene_mart) # ensembl version used
## Creating "All_attributes.txt" and "All_filters.txt" with all options:
All_attributes <- listAttributes(gene_mart)
All_filters <- listFilters(gene_mart)

write.table(All_attributes, file = "All_attributes.txt", sep = "\t",
            row.names = FALSE, quote = FALSE)
write.table(All_filters, file = "All_filters.txt", sep = "\t",
            row.names = FALSE, quote = FALSE)
# Extracting all genes within NUMTs coordinates ----
library(plyr); library(dplyr)

## Adapting coordenates to download attributes
numts_coord$coord_n <- do.call(paste, c(numts_coord[,2:4], sep = ":"))
numts_vector <-as.vector(t(numts_coord$coord_n))
id <-as.vector(t(numts_coord$id))

## Setting attributes and filters
```

```r
### Our attributes
attributes_gene = c("chromosome_name", "start_position", "end_position", "strand",
                    "hgnc_symbol", "ensembl_gene_id_version","ensembl_gene_id",
                    "transcript_count")

## Getting values: gene_results.txt (loop)

gene_results <- numeric(0)
i <- 1

for (i in 1:length(numts_vector)) {
  b<-i

  gene_results_b = getBM(attributes_gene,
                         filters = c("chromosomal_region"),
                         values = list(chromosomal_region=numts_vector[b]),
                         mart = gene_mart)

  if (length(gene_results_b[,1]) == 0) {
    gene_results <- rbind(gene_results, c(rep("", length(attributes_gene)),
                                          do.call(paste, list(numts_coord[b,1]))))
  } else {
    gene_results_b$id <- do.call(paste, list(numts_coord[b,1]))
    gene_results <- rbind(gene_results, gene_results_b)
  }

  i <- i + 1
}

gene_results[gene_results==""]  <- NA

## Reordering columns (gene_results.txt)

gene_results <- gene_results %>% dplyr::select("id", everything())

## Sorting results (gene_results.txt)
gene_results <- gene_results[order(gene_results$id,
                                   gene_results$chromosome_name,
                                   gene_results$start_position),]

## Saving the results (gene_results.txt)
write.table(gene_results, file = "gene_results.txt", sep = "\t",
            quote = FALSE, row.names = FALSE)
## Uploading intermediate documents
gene_results <- read.table("gene_results.txt", header = TRUE, sep = "\t")
summary(gene_results)
# Extracting all genes upstream and downstream from the NUMTs coordinates ----
library(plyr); library(dplyr)

  ## Indicating new coordinates
up_start <- numts_coord[3] - 1000
up_end <- numts_coord[3] - 100
down_start <- numts_coord[4] + 100
down_end <- numts_coord[4] + 1000

up_numts_coord <- data.frame(numts_coord$id,
                             numts_coord$chr,
                             up_start,
```

```
                              up_end)
down_numts_coord <- data.frame(numts_coord$id,
                               numts_coord$chr,
                               down_start,
                               down_end)

  ## Adapting coordenates to download attributes
up_numts_coord$coord_n <- do.call(paste, c(up_numts_coord[,2:4], sep = ":"))
up_numts_vector <-as.vector(t(up_numts_coord$coord_n))
down_numts_coord$coord_n <- do.call(paste, c(down_numts_coord[,2:4], sep = ":"))
down_numts_vector <-as.vector(t(down_numts_coord$coord_n))

## Getting values: up_gene_results.txt (loop)

up_gene_results <- numeric(0)
i <- 1

for (i in 1:length(up_numts_vector)) {
  b<-i

  up_gene_results_b = getBM(attributes_gene,
                        filters = c("chromosomal_region"),
                        values = list(chromosomal_region=up_numts_vector[b]),
                        mart = gene_mart)

  if (length(up_gene_results_b[,1]) == 0) {
    up_gene_results <- rbind(up_gene_results,
                             c(rep("", length(attributes_gene)),
                               do.call(paste, list(numts_coord[b,1]))))
  } else {
    up_gene_results_b$id <- do.call(paste, list(numts_coord[b,1]))
    up_gene_results <- rbind(up_gene_results, up_gene_results_b)
  }

  i <- i + 1
}

up_gene_results[up_gene_results==""]  <- NA

### Reordering columns (up_gene_results.txt)
up_gene_results <- up_gene_results %>%  dplyr::select("id", everything())

### Sorting results (up_gene_results.txt)
up_gene_results <- up_gene_results[order(up_gene_results$id,
                                         up_gene_results$chromosome_name,
                                         up_gene_results$start_position),]

### Saving the results (up_gene_results.txt)
write.table(up_gene_results, file = "up_gene_results.txt", sep = "\t",
            quote = FALSE, row.names = FALSE)


## Getting values: down_gene_results.txt (loop)

down_gene_results <- numeric(0)
i <- 1
```

```r
for (i in 1:length(down_numts_vector)) {
  b<-i

  down_gene_results_b = getBM(attributes_gene,
                        filters = c("chromosomal_region"),
                        values = list(chromosomal_region=down_numts_vector[b]),
                        mart = gene_mart)

  if (length(down_gene_results_b[,1]) == 0) {
    down_gene_results <- rbind(down_gene_results,
                            c(rep("", length(attributes_gene)),
                              do.call(paste, list(numts_coord[b,1]))))
  } else {
    down_gene_results_b$id <- do.call(paste, list(numts_coord[b,1]))
    down_gene_results <- rbind(down_gene_results, down_gene_results_b)
  }

  i <- i + 1
}

down_gene_results[down_gene_results==""]  <- NA

### Reordering columns (down_gene_results.txt)
down_gene_results <- down_gene_results %>% dplyr::select("id", everything())

### Sorting results (down_gene_results.txt)
down_gene_results <- down_gene_results[order(down_gene_results$id,
                                             down_gene_results$chromosome_name,
                                             down_gene_results$start_position),]

### Saving the results (down_gene_results.txt)
write.table(down_gene_results, file = "down_gene_results.txt", sep = "\t",
            quote = FALSE, row.names = FALSE)
## Uploading intermediate documents
up_gene_results <- read.table("up_gene_results.txt", header = TRUE, sep = "\t")
str(up_gene_results, vec.len = 1)

down_gene_results <- read.table("down_gene_results.txt", header = TRUE, sep = "\t")
str(down_gene_results, vec.len = 1)

# Extracting genes originated by NUMT insertions ----
## Total genes
TOTAL_GENES <- as.character(na.omit(unique(gene_results$ensembl_gene_id_version)))
UP_GENES <- as.character(na.omit(unique(up_gene_results$ensembl_gene_id_version)))
DOWN_GENES <- as.character(na.omit(unique(down_gene_results$ensembl_gene_id_version)))
## Common genes present in all three list of genes
## (gene_results, upstream and downstream)
library(plyr); library(dplyr)
data_joined <- dplyr::inner_join(up_gene_results, down_gene_results)

## Creating column "Localization"
data_joined$localization <- data_joined$ensembl_gene_id
data_joined$localization[!is.na(data_joined$localization)] <- "intronic"
data_joined$localization[is.na(data_joined$localization)] <- "intergenic"

## Genes in gene_results.txt but not in upper and downstream regions
int_gene_results <- anti_join(gene_results, data_joined)
```

```
#### START of exclusive from my dataset (TFM) ####

## Excluding the gene ARHGAP15

for (i in 1:nrow(int_gene_results)) {
  if (!is.na(int_gene_results$hgnc_symbol[i])) {
    if (int_gene_results$hgnc_symbol[i] == "ARHGAP15") {
      int_gene_results[i,c(2:ncol(int_gene_results))] <- NA
    } else {
      int_gene_results$hgnc_symbol[i] <- int_gene_results$hgnc_symbol[i]
    }
    i <- i + 1
  } else {
    int_gene_results$hgnc_symbol[i] <- NA
  }
}
#### END ####
## Saving genes.txt
genes <- as.character(na.omit(unique(int_gene_results$ensembl_gene_id)))

write.table(genes, file="genes.txt",col.names = F, sep="\t",quote=F,row.names=F)
## Uploading intermediate documents
genes <- read.table("genes.txt", header = FALSE, sep = "\t")

table_numts_genes <- gene_results[gene_results$ensembl_gene_id
                                  %in% genes$V1,]

chrom <- (unique(table_numts_genes[c(2,7)]))

# Number of genes per chromosome
table(chrom[1])
# GO terms (go_results.txt) ----
## Setting attributes and filters
 ### Our attributes
attributes_go = c("hgnc_symbol", "ensembl_gene_id_version",
                  "go_id", "name_1006", "definition_1006")

go_results = getBM(attributes_go,
                   filters = c("ensembl_gene_id"),
                   values = list(ensembl_gene_id=genes$V1),
                   mart = gene_mart)

go_results[go_results==""]  <- NA

go_results <- go_results[order(go_results$ensembl_gene_id_version, go_results$go_id),]

go_results <- go_results[complete.cases(go_results$name_1006), ]

write.table(go_results, file = "go_results.txt", sep = "\t",
            quote = FALSE, row.names = FALSE)
go_results <-read.table("go_results.txt", header = TRUE, sep = "\t")
go_results[c(1:4)]
## Plotting GO results
par(mfrow=c(1,2))
par(mar = c(8.5, 2.5, 2.5, 4), xpd=TRUE)

ensembl_go <- na.omit(unique(go_results[c("hgnc_symbol", "name_1006")]))
```

```r
ensembl_go <- ensembl_go[complete.cases(ensembl_go), ]

colors = c("aquamarine3","yellow2","azure3",
          "darkgoldenrod1", "lawngreen", "plum",
          "gray9","deeppink1","cornflowerblue",
          "antiquewhite3", "slategrey", "tomato")

### Bar plot
barplot(table(ensembl_go$name_1006), las=2, cex.main = 1.2,
        cex.axis = 0.7, cex = 0.7,  col = colors)


### pie chart
counts = table(ensembl_go$name_1006)  ## get counts
labs = paste(levels(ensembl_go$name_1006), counts)  ## create labels
pie(counts, labels = labs, col = colors, cex=0.5)  ## plot
legend("bottom", inset=c(0,-0.6), labs, cex=0.6, fill=colors)
# phenotype_results.txt ----
## Setting attributes and filters
### Our attributes
attributes_phenotype = c("hgnc_symbol", "ensembl_gene_id_version", "transcript_count",
                         "gene_biotype", "description")

## Getting values: phenotype_results ----

phenotype_results = getBM(attributes_phenotype,
                              filters = c("ensembl_gene_id"),
                              values = list(ensembl_gene_id=genes$V1),
                              mart = gene_mart)

# class(phenotype_results) # data.frame
phenotype_results[phenotype_results==""]  <- NA

phenotype_results <- phenotype_results[order(phenotype_results$ensembl_gene_id_version),]

write.table(phenotype_results, file = "phenotype_results.txt", sep = "\t",
            quote = FALSE, row.names = FALSE)

## Uploading intermediate documents
phenotype_results <-read.table("phenotype_results.txt",
                               header = TRUE, sep = "\t")
summary(phenotype_results)
## Plotting Gene biotype results
par(mfrow=c(1,2))
par(mar = c(6, 2.5, 2.5, 2.5), xpd=TRUE)

ensembl_biotype <- na.omit(unique(phenotype_results[c("ensembl_gene_id_version",
                                                      "gene_biotype")]))

colors = c("yellow2","orchid3","orangered3",
          "olivedrab3", "lightskyblue3", "plum")

### Bar plot
barplot(table(ensembl_biotype$gene_biotype), las=2, cex.main = 1.2,
        cex.axis = 0.7, cex = 0.7,  col = colors)

### pie chart
```

```r
counts = table(ensembl_biotype$gene_biotype)  ## get counts
labs = paste(levels(ensembl_biotype$gene_biotype), counts)  ## create labels
pie(counts, labels = labs, col = colors, cex=0.5)  ## plot
legend("bottom", inset=c(0, -0.2), labs, cex=0.6, fill=colors)
# EXPRESSION STUDY ----
## Uploading GTEx means in TPM
GTEx_mean_tpm <-
  read.table(file = params$file2, skip = 2,
             header = TRUE, sep = "\t")
## Creating mean_tpm_GTEx.txt
library(plyr); library(dplyr)

GTEx_tpm <- GTEx_mean_tpm
colnames(GTEx_tpm)[1] <- "GTEx_gene_id_version"

gene_id <- GTEx_mean_tpm$gene_id
GTEx_genes <- numeric(0)
for (i in 1:length(GTEx_mean_tpm$gene_id)){
    x <- unlist(strsplit(as.character(GTEx_mean_tpm[i,1]), split='.', fixed=TRUE))[1]
    GTEx_genes <- rbind(GTEx_genes, x)
}

GTEx_mean_tpm$gene_id <- GTEx_genes

mean_tpm_fromGTEx <- numeric(0)
for (i in 1:nrow(genes)){
    y <- subset(GTEx_mean_tpm, gene_id == genes[i,1])
    mean_tpm_fromGTEx <- rbind(mean_tpm_fromGTEx, y)
}

tissue_means <- rowMeans(mean_tpm_fromGTEx[,3:length(mean_tpm_fromGTEx)])
mean_tpm_fromGTEx$tissue_means <- tissue_means

mean_tpm_fromGTEx$sum <- rowSums(mean_tpm_fromGTEx[,3:length(mean_tpm_fromGTEx)])

mean_tpm_fromGTEx$gene_id <- as.character(mean_tpm_fromGTEx$gene_id)

mean_tpm_GTEx <- dplyr::inner_join(mean_tpm_fromGTEx,
                                   GTEx_tpm)

mean_tpm_GTEx <- mean_tpm_GTEx %>% dplyr::select("gene_id", "GTEx_gene_id_version",
                                                 everything())
colnames(mean_tpm_GTEx)[1] <- "ensembl_gene_id"
str(mean_tpm_GTEx)

# Saving results
write.table(mean_tpm_GTEx, file = "mean_tpm_GTEx.txt", sep = "\t",
            quote = FALSE, row.names = FALSE)
## Uploading intermediate documents
mean_tpm_GTEx <- read.table("mean_tpm_GTEx.txt", header = TRUE,
                                   sep = "\t", dec = ".")
# subset_expressed.txt ----
## Creating subset of genes with >= 0.5 TPM (expressed)
GTEx <- mean_tpm_GTEx
subset_expressed <- subset(GTEx, GTEx[4] >= 0.5 | GTEx[5] >= 0.5 | GTEx[6] >= 0.5 |
                             GTEx[7] >= 0.5 | GTEx[8] >= 0.5 | GTEx[9] >= 0.5 |
                             GTEx[10] >= 0.5 |GTEx[11] >= 0.5 | GTEx[12] >= 0.5 |
```

```
                              GTEx[13] >= 0.5 | GTEx[14] >= 0.5 | GTEx[15] >= 0.5 |
                              GTEx[16] >= 0.5 | GTEx[17] >= 0.5 | GTEx[18] >= 0.5 |
                              GTEx[19] >= 0.5 | GTEx[20] >= 0.5 | GTEx[21] >= 0.5 |
                              GTEx[22] >= 0.5 | GTEx[23] >= 0.5 | GTEx[24] >= 0.5 |
                              GTEx[25] >= 0.5 | GTEx[26] >= 0.5 | GTEx[27] >= 0.5 |
                              GTEx[28] >= 0.5 | GTEx[29] >= 0.5 | GTEx[30] >= 0.5 |
                              GTEx[31] >= 0.5 | GTEx[32] >= 0.5 | GTEx[33] >= 0.5 |
                              GTEx[34] >= 0.5 | GTEx[35] >= 0.5 | GTEx[36] >= 0.5 |
                              GTEx[37] >= 0.5 | GTEx[38] >= 0.5 | GTEx[39] >= 0.5 |
                              GTEx[40] >= 0.5 | GTEx[41] >= 0.5 | GTEx[42] >= 0.5 |
                              GTEx[43] >= 0.5 | GTEx[44] >= 0.5 | GTEx[45] >= 0.5 |
                              GTEx[46] >= 0.5 | GTEx[47] >= 0.5 | GTEx[48] >= 0.5 |
                              GTEx[49] >= 0.5 | GTEx[50] >= 0.5 | GTEx[51] >= 0.5 |
                              GTEx[52] >= 0.5 | GTEx[53] >= 0.5 | GTEx[54] >= 0.5 |
                              GTEx[55] >= 0.5 | GTEx[56] >= 0.5 | GTEx[57] >= 0.5 )

## saving results
write.table(subset_expressed, file = "subset_expressed.txt", sep = "\t",
            quote = FALSE, row.names = FALSE)
## Uploading intermediate documents
subset_expressed <- read.table("subset_expressed.txt", header = TRUE,
                               sep = "\t", dec = ".")
summary(subset_expressed[c(4:ncol(subset_expressed))])
## Heatmap of all expressed genes.
library(gplots)
par(oma=c(10,4,4,2))
subset_mean_tpm <- subset_expressed
subset_mean_tpm[1] <- NULL
rownames(subset_mean_tpm) <- subset_mean_tpm$Description
subset_mean_tpm[1] <- NULL
subset_mean_tpm[1] <- NULL

heatmap.2(data.matrix(subset_mean_tpm[1:53]), trace='none', scale = "row",
          cexRow=0.6, cexCol = 0.6)

## Heatmap of all expressed genes.
library(gplots)
par(oma=c(10,4,4,2))
subset_mean_tpm <- subset_expressed
subset_mean_tpm[1] <- NULL
rownames(subset_mean_tpm) <- subset_mean_tpm$Description
subset_mean_tpm[1] <- NULL
subset_mean_tpm[1] <- NULL

heatmap.2(data.matrix(subset_mean_tpm[1:53]), trace='none', scale = "column",
          cexRow=0.6, cexCol = 0.6)

## Graphical representation: expression profile of
## hight expressed genes (>=1000 TMP)

par(mar = c(10.5, 4, 4, 7.5), xpd=TRUE)
GTEx1 <- subset_mean_tpm
subset_mean_tpm2 <- subset(GTEx1, GTEx1[1] >= 1000.0 |GTEx1[2] >= 1000.0 |
                              GTEx1[3] >= 1000.0 | GTEx1[4] >= 1000.0 |
                              GTEx1[5] >= 1000.0 | GTEx1[6] >= 1000.0 |
                              GTEx1[7] >= 1000.0 | GTEx1[8] >= 1000.0 |
                              GTEx1[9] >= 1000.0 | GTEx1[10] >= 1000.0 |
```

```
                              GTEx1[11] >= 1000.0 | GTEx1[12] >= 1000.0 |
                              GTEx1[13] >= 1000.0 | GTEx1[14] >= 1000.0 |
                              GTEx1[15] >= 1000.0 | GTEx1[16] >= 1000.0 |
                              GTEx1[17] >= 1000.0 | GTEx1[18] >= 1000.0 |
                              GTEx1[19] >= 1000.0 | GTEx1[20] >= 1000.0 |
                              GTEx1[21] >= 1000.0 | GTEx1[22] >= 1000.0 |
                              GTEx1[23] >= 1000.0 | GTEx1[24] >= 1000.0 |
                              GTEx1[25] >= 1000.0 | GTEx1[26] >= 1000.0 |
                              GTEx1[27] >= 1000.0 | GTEx1[28] >= 1000.0 |
                              GTEx1[29] >= 1000.0 | GTEx1[30] >= 1000.0 |
                              GTEx1[31] >= 1000.0 | GTEx1[32] >= 1000.0 |
                              GTEx1[33] >= 1000.0 | GTEx1[34] >= 1000.0 |
                              GTEx1[35] >= 1000.0 | GTEx1[36] >= 1000.0 |
                              GTEx1[37] >= 1000.0 | GTEx1[38] >= 1000.0 |
                              GTEx1[39] >= 1000.0 | GTEx1[40] >= 1000.0 |
                              GTEx1[41] >= 1000.0 | GTEx1[42] >= 1000.0 |
                              GTEx1[43] >= 1000.0 | GTEx1[44] >= 1000.0 |
                              GTEx1[45] >= 1000.0 | GTEx1[46] >= 1000.0 |
                              GTEx1[47] >= 1000.0 | GTEx1[48] >= 1000.0 |
                              GTEx1[49] >= 1000.0 | GTEx1[50] >= 1000.0 |
                              GTEx1[51] >= 1000.0 | GTEx1[52] >= 1000.0 |
                              GTEx1[53] >= 1000.0 )
matplot(t(data.matrix(subset_mean_tpm2[1:53])), type = "b",
        col = c(1:ncol(subset_mean_tpm2)),
        cex.main = 1, cex.lab = 0.8,    ylab = "TPM", pch=c(15:18,21:25),
        main = "TPM/Tissue", axes = FALSE)
axis(2, cex.axis=0.7)
axis(side=1,at=1:ncol(subset_mean_tpm2[1:53]), cex.axis=0.6, las = 2,
     labels=colnames(subset_mean_tpm2[1:53]))

legend("right", inset=c(-0.25, 1), legend=rownames(subset_mean_tpm2[1:53]),
       col=c(1:ncol(subset_mean_tpm2)),pch= c(15:18,21:25),
       cex = 0.6, bg= ("white"), horiz=F)

## Graphical representation: expression profile of
## medium expressed genes (between 10 and 1000 TMP)

subset_mean_tpm3 <- subset(GTEx1, GTEx1[1] >= 10.0 |GTEx1[2] >= 10.0 |
                              GTEx1[3] >= 10.0 | GTEx1[4] >= 10.0 |
                              GTEx1[5] >= 10.0 | GTEx1[6] >= 10.0 |
                              GTEx1[7] >= 10.0 | GTEx1[8] >= 10.0 |
                              GTEx1[9] >= 10.0 | GTEx1[10] >= 10.0 |
                              GTEx1[11] >= 10.0 | GTEx1[12] >= 10.0 |
                              GTEx1[13] >= 10.0 | GTEx1[14] >= 10.0 |
                              GTEx1[15] >= 10.0 | GTEx1[16] >= 10.0 |
                              GTEx1[17] >= 10.0 | GTEx1[18] >= 10.0 |
                              GTEx1[19] >= 10.0 | GTEx1[20] >= 10.0 |
                              GTEx1[21] >= 10.0 | GTEx1[22] >= 10.0 |
                              GTEx1[23] >= 10.0 | GTEx1[24] >= 10.0 |
                              GTEx1[25] >= 10.0 | GTEx1[26] >= 10.0 |
                              GTEx1[27] >= 10.0 | GTEx1[28] >= 10.0 |
                              GTEx1[29] >= 10.0 | GTEx1[30] >= 10.0 |
                              GTEx1[31] >= 10.0 | GTEx1[32] >= 10.0 |
                              GTEx1[33] >= 10.0 | GTEx1[34] >= 10.0 |
                              GTEx1[35] >= 10.0 | GTEx1[36] >= 10.0 |
                              GTEx1[37] >= 10.0 | GTEx1[38] >= 10.0 |
                              GTEx1[39] >= 10.0 | GTEx1[40] >= 10.0 |
```

```
                              GTEx1[41] >= 10.0 | GTEx1[42] >= 10.0 |
                              GTEx1[43] >= 10.0 | GTEx1[44] >= 10.0 |
                              GTEx1[45] >= 10.0 | GTEx1[46] >= 10.0 |
                              GTEx1[47] >= 10.0 | GTEx1[48] >= 10.0 |
                              GTEx1[49] >= 10.0 | GTEx1[50] >= 10.0 |
                              GTEx1[51] >= 10.0 | GTEx1[52] >= 10.0 |
                              GTEx1[53] >= 10.0 )
subset_mean_tpm3 <- subset_mean_tpm3[!rownames(subset_mean_tpm3) %in%
                                     rownames(subset_mean_tpm2), ]

par(mar = c(10.5, 4, 4, 7.5), xpd=TRUE)
matplot(t(data.matrix(subset_mean_tpm3[1:53])), type = "b",
        col = c(1:ncol(subset_mean_tpm3)),
        cex.main = 1, cex.lab = 0.8,    ylab = "TPM", pch=c(15:18,21:25),
        main = "TPM/Tissue", axes = FALSE)
axis(2, cex.axis=0.7)
axis(side=1,at=1:ncol(subset_mean_tpm3[1:53]), cex.axis=0.6, las = 2,
     labels=colnames(subset_mean_tpm3[1:53]))

legend("right", inset=c(-0.25, 1), legend=rownames(subset_mean_tpm3[1:53]),
       col=c(1:ncol(subset_mean_tpm3)),pch= c(15:18,21:25), cex = 0.6,
       bg= ("white"), horiz=F)

# FINAL TABLE: FINAL_OUTPUT_TABLE.txt ----
library(plyr); library(dplyr)

## Cheking genes and creating table
table_numts_genes <- gene_results[gene_results$ensembl_gene_id
                                  %in% genes$V1,]

str(table_numts_genes)
length(unique(table_numts_genes$ensembl_gene_id))
summary(table_numts_genes)

table_numts_genes <- dplyr::full_join(numts_coord,
                         table_numts_genes)

table_numts_genes <- dplyr::full_join(data_joined[c("id", "localization")],
                         table_numts_genes)
table_numts_genes$localization[is.na(table_numts_genes$localization)] <- "partial_gene"
table_numts_genes <- dplyr::full_join(phenotype_results[c("ensembl_gene_id_version",
                                          "gene_biotype", "description")],
                                table_numts_genes)

table_numts_go <- dplyr::full_join(go_results[c("ensembl_gene_id_version",
                                       "name_1006")],
                         table_numts_genes)

table_numts_exp <- dplyr::full_join(mean_tpm_GTEx,
                         table_numts_go)

table_numts <- table_numts_exp %>% dplyr::select("id", "localization", "chr",
                                                 "start_n", "end_n", "mt",
                                                 "start_mt", "end_mt",
                                                 "hgnc_symbol", "Description",
                                                 "gene_biotype", "name_1006",
                                                 "ensembl_gene_id_version",
```

```
                                                    "transcript_count",
                                                    "tissue_means", "sum",
                                                    everything())

## Removing columns
table_numts$ensembl_gene_id = NULL

## Sorting results (gene_results.txt) ----

table_numts <- table_numts[order(table_numts$id,
                                 table_numts$chr,
                                 table_numts$start_n,
                                 table_numts$start_position),]

table_numts <- table_numts[!duplicated(table_numts), ]

## Saving results
write.table(table_numts, file="FINAL_OUTPUT_TABLE.txt",
            sep="\t",quote=F,row.names=F)
## Uploading intermediate documents
table_numts <- read.table("FINAL_OUTPUT_TABLE.txt", header = TRUE, sep = "\t")
## Showing 6 first data from FINAL TABLE "FINAL_OUTPUT_TABLE.txt"
head(table_numts)
library(plyr); library(dplyr)

only_expressed <- table_numts[table_numts$ensembl_gene_id
                                 %in% subset_expressed$GTEx_gene_id_version,]
## Plotting Gene biotype results
par(mfrow=c(1,2))
par(mar = c(6, 2.5, 2.5, 2.5), xpd=TRUE)

ensembl_biotype_ex <- na.omit(unique(only_expressed[c("ensembl_gene_id_version",
                                                      "gene_biotype")]))

ensembl_go_ex <- na.omit(unique(only_expressed[c("ensembl_gene_id_version",
                                                      "name_1006")]))
colors = c("yellow2","orchid3","orangered3",
          "olivedrab3", "lightskyblue3", "plum")

### pie chart
counts = table(ensembl_biotype_ex$gene_biotype)  ## get counts
labs = paste(levels(ensembl_biotype$gene_biotype), counts)  ## create labels
pie(counts, labels = labs, col = colors, cex=0.5)  ## plot
legend("bottom", inset=c(0, -0.2), labs, cex=0.6, fill=colors)

## Plotting GO results

ensembl_go <- ensembl_go_ex[complete.cases(ensembl_go_ex), ]

colors = c("aquamarine3","yellow2","azure3",
          "darkgoldenrod1", "lawngreen", "plum",
          "gray9","deeppink1","cornflowerblue",
          "antiquewhite3", "slategrey", "tomato")

### pie chart
counts = table(ensembl_go$name_1006)  ## get counts
labs = paste(levels(ensembl_go$name_1006), counts)  ## create labels
pie(counts, labels = labs, col = colors, cex=0.5)  ## plot
```

```
legend("bottom", inset=c(0, -0.2), labs, cex=0.6, fill=colors)


######## PART 2: Additional scripts for TFM ##########

#install.packages("overlap")
#install.packages("DescTools")
library(DescTools)
library(plyr)
library(dplyr)
library(biomaRt)

# First, we will check general data from our analysis:
table <- gene_results[gene_results$ensembl_gene_id
                                  %in% genes$V1,]
# Genes per NUMT
numt_t <- table(table[1])

# NUMTS WITHOUT GENES
sum(numt_t == 0)

x <- numeric(0)
# Loop for the rest:
for (i in 0:length(unique(numt_t))){
x[i]  = paste("NUMTs containing ", i, " genes:",  sum(numt_t == i))
 }
x

# NUMTs per gene
numt_g <- table(table_numts_genes[7])

#Genes deleted after filtering:
sum(numt_g == 0)

y <- numeric(0)
# Loop for the rest:
for (j in 0:length(unique(numt_g))){
y[j]  = paste("Genes included in ", j, " NUMTs:",  sum(numt_g == j))
 }
y

# File 12: "gene_result_mt.txt" -----------
# Input file ----
numts_coord <- read.csv("NUMTs_coord.csv", sep = ",", header = TRUE)
numts_coord$coord_mt <- do.call(paste, c(numts_coord[,5:7], sep = ":"))
numts_vector_mt <-as.vector(t(numts_coord$coord_mt))

attributes_mt = c("chromosome_name", "start_position", "end_position", "strand",
                  "hgnc_symbol", "ensembl_gene_id", "transcript_count", "gene_biotype")
gene_results_mt <- numeric(0)
i <- 1

for (i in 1:length(numts_vector_mt)) {
  b<-i

  gene_results_b = getBM(attributes_mt,
                         filters = c("chromosomal_region"),
```

```
                          values = list(chromosomal_region=numts_vector_mt[b]),
                          mart = gene_mart)

  if (length(gene_results_b[,1]) == 0) {
    gene_results_mt <- rbind(gene_results_mt, c(rep("", length(attributes_mt)),
                                                do.call(paste, list(numts_coord[b,1]))))
  } else {
    gene_results_b$id <- do.call(paste, list(numts_coord[b,1]))
    gene_results_mt <- rbind(gene_results_mt, gene_results_b)
  }

  i <- i + 1
}

str(gene_results_mt)
class(gene_results_mt) # data.frame

gene_results_mt[gene_results_mt==""]  <- NA

## Reordering columns ----

gene_mt <- gene_results_mt %>% dplyr::select("id", "hgnc_symbol", everything())

## Sorting results  ----
gene_results_mt <- gene_mt[order(gene_results_mt$id,
                                 gene_results_mt$start_position),]
head(gene_results_mt)
ncol(gene_results_mt) # columns: 9
nrow(gene_results_mt) # rows: 3954
length(unique(gene_results_mt$hgnc_symbol)) # num. mito genes: 38

## Saving the results (gene_results_mt.txt) ----
write.table(gene_results_mt, file = "gene_results_mt.txt", sep = "\t",
            quote = FALSE, row.names = FALSE)

##############################################################

# IDENTIFICATION OF CORRESPONDING MITOCHONDRIAL GENE FOR EACH NEW
# NUCLEAR GENE WITHIN NUMTs
## Ordering
table_mito_genes <- full_join(numts_coord[c("id", "mt",
                                            "start_mt", "end_mt")],
                              gene_results_mt)

table_mito <- table_mito_genes %>% dplyr::select("id", "mt", "start_mt", "end_mt",
                                                 "hgnc_symbol", "ensembl_gene_id",
                                                 "transcript_count", "gene_biotype",
                                                 everything())


# CONDITION:
## if gene is strand is +1:
### START: (gene start - NUMT start) and END: (gene end - NUMT start)
## if gene is strand -1:
###START: (NUMT end - gene end) and END: (NUMT end - gene start)
str(table_mito)
table_mito$start_mt <- as.numeric(table_mito$start_mt)
```

```r
table_mito$start_position <- as.numeric(table_mito$start_position)
table_mito$end_mt <- as.numeric(table_mito$end_mt)
table_mito$end_position <- as.numeric(table_mito$end_position)

table_mito$start_ref = NULL
table_mito$end_ref = NULL
for (i in 1:nrow(table_mito)) {
  if (!is.na(table_mito$strand[i])) {
    if (table_mito$strand[1] == 1) {
      table_mito$start_ref[i] <- table_mito$start_position[i] - table_mito$start_mt[i]
      table_mito$end_ref[i] <- table_mito$end_position[i] - table_mito$start_mt[i]
      } else {
        table_mito$start_ref[i] <- table_mito$end_mt[i] - table_mito$end_position[i]
        table_mito$end_ref[i] <- table_mito$end_mt[i] - table_mito$start_position[i]
        }
    i <- i + 1
  } else {
    table_mito$start_ref[i] <- NA
    table_mito$end_ref[i] <- NA
    }
}
table_mito$start_ref <- as.numeric(table_mito$start_ref)
table_mito$end_ref <- as.numeric(table_mito$end_ref)
str(table_mito)
head(table_mito)
table_mito$from_numt <- NULL
table_mito$from_numt <-  substr(table_mito$id, 6,11)
table_mito$from_numt <- gsub("\\.", "", table_mito$from_numt)
table_mito$from_numt <- gsub("\\X", "23", table_mito$from_numt)
table_mito$from_numt <- gsub("\\Y", "24", table_mito$from_numt)
table_mito$from_numt <- as.numeric(table_mito$from_numt)

# In reference interval, replace negative numbers by 0
table_mito$start_ref[table_mito$start_ref<0] <- 0
table_mito$end_ref[table_mito$end_ref<0] <- 0

max(table_mito$start_ref,na.rm=T)
max(table_mito$end_ref,na.rm=T)

# Creating a exclusive numerical code ("comparable coordinates")
# Exclusive for each NUMT to compare mitocondrial and nuclar genes
# included in same NUMT

table_mito$from_numt <- table_mito$from_numt * 100000
table_mito$start_mtnumt <- table_mito$start_ref + table_mito$from_numt
table_mito$end_mtnumt <- table_mito$end_ref + table_mito$from_numt

mt_genes_coord <- data.frame("mt_hgnc_symbol" = table_mito$hgnc_symbol,
                             "mt_start_numt" = table_mito$start_mtnumt,
                             "mt_end_numt" = table_mito$end_mtnumt)
head(table_mito)
ncol(table_mito)
nrow(table_mito)

# Same process with nuclear genes ---------------

table_numts <- read.table("FINAL_OUTPUT_TABLE.txt", header = TRUE, sep = "\t")
```

```r
length(duplicated(table_numts)[duplicated(table_numts)==TRUE])
head(table_numts)
str(table_numts)

# Removing duplicated rows
table_numts <- table_numts[!duplicated(table_numts), ]
table_numts$start_n <- as.numeric(table_numts$start_n)
table_numts$start_position <- as.numeric(table_numts$start_position)
table_numts$end_n <- as.numeric(table_numts$end_n)
table_numts$end_position <- as.numeric(table_numts$end_position)

table_numts$start_ref = NULL
table_numts$end_ref = NULL
for (i in 1:nrow(table_numts)) {
  if (!is.na(table_numts$strand[i])) {
    if (table_numts$strand[1] == 1) {
      table_numts$start_ref[i] <- table_numts$start_position[i] - table_numts$start_n[i]
      table_numts$end_ref[i] <- table_numts$end_position[i] - table_numts$start_n[i]
    } else {
      table_numts$start_ref[i] <- table_numts$end_n[i] - table_numts$end_position[i]
      table_numts$end_ref[i] <- table_numts$end_n[i] - table_numts$start_position[i]
    }
    i <- i + 1
  } else {
    table_numts$start_ref[i] <- NA
    table_numts$end_ref[i] <- NA
  }
}
table_numts$start_ref <- as.numeric(table_numts$start_ref)
table_numts$end_ref <- as.numeric(table_numts$end_ref)

str(table_numts)
head(table_numts)
table_numts$from_numt <- NULL
table_numts$from_numt <-  substr(table_numts$id, 6,11)
table_numts$from_numt <- gsub("\\.", "", table_numts$from_numt)
table_numts$from_numt <- gsub("\\X", "23", table_numts$from_numt)
table_numts$from_numt <- gsub("\\Y", "24", table_numts$from_numt)
table_numts$from_numt <- as.numeric(table_numts$from_numt)

# In reference interval, replace negative numbers by 0
table_numts$start_ref[table_numts$start_ref<0] <- 0
table_numts$end_ref[table_numts$end_ref<0] <- 0

max(table_numts$start_ref,na.rm=T)
max(table_numts$end_ref,na.rm=T)

table_numts$from_numt <- table_numts$from_numt * 100000
table_numts$start_numt <- table_numts$start_ref + table_numts$from_numt
table_numts$end_numt <- table_numts$end_ref + table_numts$from_numt

n_genes_coord <- data.frame("n_hgnc_symbol" = table_numts$hgnc_symbol,
                            "n_start_numt" = table_numts$start_numt,
                            "n_end_numt" = table_numts$end_numt)
head((mt_genes_coord))
head((n_genes_coord))
```

```r
c(n_genes_coord[1,2], n_genes_coord[1,3]) %overlaps%
  c(mt_genes_coord[1,2], mt_genes_coord[1,3])

c(n_genes_coord[1,2], n_genes_coord[1,3]) %overlaps%
  c(mt_genes_coord[1,2], mt_genes_coord[1,3])

# Searching for overlaping
i <- 1
b <- 1
overlaping <- NULL
for (i in 1:nrow(n_genes_coord)) for (b in 1:nrow(mt_genes_coord))
  if (!is.na(n_genes_coord$n_start_numt[i] & n_genes_coord$n_end_numt[i] &
             mt_genes_coord$mt_start_numt[b] & mt_genes_coord$mt_end_numt[b])){
    if (c(n_genes_coord$n_start_numt[i], n_genes_coord$n_end_numt[i]) %overlaps%
        c(mt_genes_coord$mt_start_numt[b],
          mt_genes_coord$mt_end_numt[b]))
      {
      overlaping_b <- data.frame("n_hgnc_symbol" = n_genes_coord$n_hgnc_symbol[i],
                                 "mt_hgnc_symbol" = mt_genes_coord$mt_hgnc_symbol[b],
                                 "n_start_numt" = n_genes_coord$n_start_numt[i],
                                 "n_end_numt" = n_genes_coord$n_end_numt[i],
                                 "mt_start_numt" = mt_genes_coord$mt_start_numt[b],
                                 "mt_end_numt" = mt_genes_coord$mt_end_numt[b])
      overlaping <- rbind(overlaping, overlaping_b)

    }
  }


overlaping <- overlaping[!duplicated(overlaping), ]
head(overlaping)

overlaping$n_lenght <- overlaping$n_end_numt - overlaping$n_start_numt
overlaping$mt_lenght <- overlaping$mt_end_numt - overlaping$mt_start_numt

for (i in 1:nrow(overlaping)) {
overlaping$ov[i] <- (min(c(overlaping$n_end_numt[i],
                           overlaping$mt_end_numt[i]))
                     - max(c(overlaping$n_start_numt[i],
                             overlaping$mt_start_numt[i])))
}

overlaping$percent_n <- overlaping$ov/overlaping$n_lenght
overlaping$percent_mt <- overlaping$ov/overlaping$mt_lenght
head(overlaping)

# To focuss on nuclear genes that are mainly originated from mitochondrial genes
# we filter the output for at least, 70% of representation of mitochondrial gene
# or 70% of nuclear gene originated from a mitochondrial gene

total_overlaping <- overlaping

# CREATING TABLE total_
write.table(total_overlaping, file = "total_overlaping.txt", sep = "\t",
            quote = FALSE, row.names = FALSE)

overlaping <- subset(overlaping, percent_n >= 0.7 |
```

```
                          percent_mt >= 0.7 )


# ORDERING DATA ----

head(table_mito)
mito <- table_mito[c(1,2,3,4,5,6,8,10,11,12,13,14,16,17)]
head(mito)
colnames(mito)[5] <- "mt_hgnc_symbol"
colnames(mito)[6] <- "mt_ensembl_gene_id"
colnames(mito)[7] <- "mt_gene_biotype"
colnames(mito)[8] <- "mt_start_position"
colnames(mito)[9] <- "mt_end_position"
colnames(mito)[10] <- "mt_strand"
colnames(mito)[11] <- "mt_start_ref"
colnames(mito)[12] <- "mt_end_ref"
colnames(mito)[13] <- "mt_start_numt"
colnames(mito)[14] <- "mt_end_numt"

head(mito)
head(overlaping)

nrow(mito)
nrow(overlaping)
table_overlap <- dplyr::inner_join(mito, overlaping)

head(table_numts)

head(data)
data <- table_numts[c(1:13,72:76,78,79,14:70)]
head(data)

colnames(data)[9] <- "n_hgnc_symbol"
colnames(data)[11] <- "n_gene_biotype"
colnames(data)[12] <- "GO_term"
colnames(data)[14] <- "n_start_position"
colnames(data)[15] <- "n_end_position"
colnames(data)[16] <- "n_strand"
colnames(data)[17] <- "n_start_ref"
colnames(data)[18] <- "n_end_ref"
colnames(data)[19] <- "n_start_numt"
colnames(data)[20] <- "n_end_numt"



data <- data[!duplicated(data), ]
nrow(data)

all_data <- dplyr::full_join(data, table_overlap)
all_data <- all_data[!duplicated(all_data), ]
nrow(all_data)

# Ordering

all_data$n_start_numt <- NULL
all_data$n_end_numt <- NULL
all_data$mt_start_numt <- NULL
```

```r
all_data$mt_end_numt <- NULL
all_data$NUMT_size <- all_data$end_n  - all_data$start_n
all_data <- all_data %>% dplyr::select("id", "localization",
                                       "chr", "start_n", "end_n",
                                       "NUMT_size",
                                       "mt", "start_mt", "end_mt",
                                       "Description", "n_gene_biotype",
                                       "mt_hgnc_symbol","mt_gene_biotype",
                                       "ensembl_gene_id_version",

                                       "n_strand",
                                       "n_start_position", "n_end_position",
                                       "mt_ensembl_gene_id",
                                       "mt_strand",
                                       "mt_start_position", "mt_end_position",
                                       "n_start_ref", "n_end_ref",
                                       "mt_start_ref", "mt_end_ref",
                                       "n_lenght", "mt_lenght",
                                       "ov", "percent_n", "percent_mt",
                                       "n_hgnc_symbol",
                                       everything())

# # # # # # CREATING DEFINITIVE TABLE "all_data_70.txt" # # # # # # #

write.table(all_data, file = "all_data_70.txt", sep = "\t",
            quote = FALSE, row.names = FALSE)


##############################################################

# # # # # # MITOCHONDRIAL GENES EXPRESSION # # # # # # #

# Saving complete list of mitochondrial genes ( "mito_genes.txt")

gene_results_mt$ensembl_gene_id
mito_genes <- as.character(na.omit(unique(gene_results_mt$ensembl_gene_id)))
length(mito_genes)
write.table(mito_genes, file="mito_genes.txt",col.names = F,
            sep="\t",quote=F,row.names=F)

# Generating expression data "mean_tpm_GTExMITO.txt"

genes <- read.table("mito_genes.txt", header = FALSE, sep = "\t")
GTEx_mean_tpm <-
  read.table("GTEx_Analysis_2016-01-15_v7_RNASeQCv1.1.8_gene_median_tpm.gct",
             skip = 2, header = TRUE, sep = "\t")

library(plyr); library(dplyr)

GTEx_tpm <- GTEx_mean_tpm
colnames(GTEx_tpm)[1] <- "GTEx_gene_id_version"

gene_id <- GTEx_mean_tpm$gene_id
GTEx_genes <- numeric(0)
for (i in 1:length(GTEx_mean_tpm$gene_id)){
    x <- unlist(strsplit(as.character(GTEx_mean_tpm[i,1]), split='.', fixed=TRUE))[1]
    GTEx_genes <- rbind(GTEx_genes, x)
```

```r
}

GTEx_mean_tpm$gene_id <- GTEx_genes

mean_tpm_fromGTEx <- numeric(0)
for (i in 1:nrow(genes)){
    y <- subset(GTEx_mean_tpm, gene_id == genes[i,1])
    mean_tpm_fromGTEx <- rbind(mean_tpm_fromGTEx, y)
}

tissue_means <- rowMeans(mean_tpm_fromGTEx[,3:length(mean_tpm_fromGTEx)])
mean_tpm_fromGTEx$tissue_means <- tissue_means

mean_tpm_fromGTEx$sum <- rowSums(mean_tpm_fromGTEx[,3:length(mean_tpm_fromGTEx)])

mean_tpm_fromGTEx$gene_id <- as.character(mean_tpm_fromGTEx$gene_id)

mean_tpm_GTExMITO <- inner_join(mean_tpm_fromGTEx,
                                GTEx_tpm)

mean_tpm_GTExMITO <- mean_tpm_GTExMITO %>%
  select("gene_id", "GTEx_gene_id_version", everything())

colnames(mean_tpm_GTExMITO)[1] <- "ensembl_gene_id"
str(mean_tpm_GTExMITO)

# Saving results

write.table(mean_tpm_GTExMITO, file = "mean_tpm_GTExMITO.txt",
            sep = "\t", quote = FALSE, row.names = FALSE)

mean_tpm_GTExMITO <- read.table("mean_tpm_GTExMITO.txt", header = TRUE,
                                sep = "\t", dec = ".")
nrow(genes)
ncol(genes)
nrow(mean_tpm_GTExMITO)
ncol(mean_tpm_GTExMITO)

head(mean_tpm_GTExMITO[c(1,2,3)])

GTEx <- mean_tpm_GTExMITO
subset_expressed <- GTEx

# fig.cap= "Heat map of all expressed genes."
library(gplots)
par(oma=c(10,4,4,2))
subset_mean_tpm <- subset_expressed
subset_mean_tpm[1] <- NULL
rownames(subset_mean_tpm) <- subset_mean_tpm$Description
subset_mean_tpm[1] <- NULL
subset_mean_tpm[1] <- NULL

heatmap.2(data.matrix(subset_mean_tpm[1:53]), trace='none', scale = "row",
          cexRow=0.6, cexCol = 0.6)

# fig.cap= "Heat map of all expressed genes."
library(gplots)
```

```r
par(oma=c(10,4,4,2))
subset_mean_tpm <- subset_expressed
subset_mean_tpm[1] <- NULL
rownames(subset_mean_tpm) <- subset_mean_tpm$Description
subset_mean_tpm[1] <- NULL
subset_mean_tpm[1] <- NULL

heatmap.2(data.matrix(subset_mean_tpm[1:53]),
          trace='none', scale = "column",
          cexRow=0.6, cexCol = 0.6)


# # # # # # # # # # # # sessionInf() # # # # # # # # # # # # #
devtools::session_info()
```

```
##  setting  value
##  version  R version 3.4.4 (2018-03-15)
##  system   x86_64, linux-gnu
##  ui       X11
##  language en_US
##  collate  en_US.UTF-8
##  tz       Europe/Madrid
##  date     2018-06-05
##
##  package       * version  date       source
##  AnnotationDbi   1.40.0   2018-04-19 Bioconductor
##  assertthat      0.2.0    2017-04-11 CRAN (R 3.4.4)
##  backports       1.1.2    2017-12-13 CRAN (R 3.4.4)
##  base          * 3.4.4    2018-03-16 local
##  bindr           0.1.1    2018-03-13 CRAN (R 3.4.4)
##  bindrcpp        0.2.2    2018-03-29 CRAN (R 3.4.4)
##  Biobase         2.38.0   2018-04-25 Bioconductor
##  BiocGenerics    0.24.0   2018-04-19 Bioconductor
##  BiocInstaller * 1.28.0   2018-04-19 Bioconductor
##  biomaRt       * 2.34.2   2018-05-20 Bioconductor
##  bit             1.1-13   2018-05-15 CRAN (R 3.4.4)
##  bit64           0.9-7    2017-05-08 CRAN (R 3.4.4)
##  bitops          1.0-6    2013-08-17 CRAN (R 3.4.4)
##  blob            1.1.1    2018-03-25 CRAN (R 3.4.4)
##  caTools         1.17.1   2014-09-10 CRAN (R 3.4.4)
##  compiler        3.4.4    2018-03-16 local
##  datasets      * 3.4.4    2018-03-16 local
##  DBI             1.0.0    2018-05-02 CRAN (R 3.4.4)
##  devtools        1.13.5   2018-02-18 CRAN (R 3.4.4)
##  digest          0.6.15   2018-01-28 CRAN (R 3.4.4)
##  dplyr         * 0.7.5    2018-05-19 CRAN (R 3.4.4)
##  evaluate        0.10.1   2017-06-24 CRAN (R 3.4.4)
##  gdata           2.18.0   2017-06-06 CRAN (R 3.4.4)
##  glue            1.2.0    2017-10-29 CRAN (R 3.4.4)
##  gplots        * 3.0.1    2016-03-30 CRAN (R 3.4.4)
##  graphics      * 3.4.4    2018-03-16 local
##  grDevices     * 3.4.4    2018-03-16 local
##  gtools          3.5.0    2015-05-29 CRAN (R 3.4.4)
##  htmltools       0.3.6    2017-04-28 CRAN (R 3.4.4)
##  httr            1.3.1    2017-08-20 CRAN (R 3.4.4)
##  IRanges         2.12.0   2018-04-19 Bioconductor
##  KernSmooth      2.23-15  2015-06-29 CRAN (R 3.4.0)
##  knitr           1.20     2018-02-20 CRAN (R 3.4.4)
```

```
##  magrittr        1.5      2014-11-22 CRAN (R 3.4.4)
##  memoise         1.1.0    2017-04-21 CRAN (R 3.4.4)
##  methods       * 3.4.4    2018-03-16 local
##  parallel        3.4.4    2018-03-16 local
##  pillar          1.2.2    2018-04-26 CRAN (R 3.4.4)
##  pkgconfig       2.0.1    2017-03-21 CRAN (R 3.4.4)
##  plyr          * 1.8.4    2016-06-08 CRAN (R 3.4.4)
##  prettyunits     1.0.2    2015-07-13 CRAN (R 3.4.4)
##  progress        1.1.2    2016-12-14 CRAN (R 3.4.4)
##  purrr           0.2.4    2017-10-18 CRAN (R 3.4.4)
##  R6              2.2.2    2017-06-17 CRAN (R 3.4.4)
##  Rcpp            0.12.17  2018-05-18 CRAN (R 3.4.4)
##  RCurl           1.95-4.7 2015-06-30 CRAN (R 3.2.2)
##  rlang           0.2.0    2018-02-20 CRAN (R 3.4.4)
##  rmarkdown       1.9      2018-03-01 CRAN (R 3.4.4)
##  rprojroot       1.3-2    2018-01-03 CRAN (R 3.4.4)
##  RSQLite         2.1.1    2018-05-06 CRAN (R 3.4.4)
##  S4Vectors       0.16.0   2018-04-19 Bioconductor
##  stats         * 3.4.4    2018-03-16 local
##  stats4          3.4.4    2018-03-16 local
##  stringi         1.2.2    2018-05-02 CRAN (R 3.4.4)
##  stringr         1.3.1    2018-05-10 CRAN (R 3.4.4)
##  tibble          1.4.2    2018-01-22 CRAN (R 3.4.4)
##  tidyselect      0.2.4    2018-02-26 CRAN (R 3.4.4)
##  tools           3.4.4    2018-03-16 local
##  utils         * 3.4.4    2018-03-16 local
##  withr           2.1.2    2018-03-15 CRAN (R 3.4.4)
##  XML             3.98-1.3 2015-06-30 CRAN (R 3.2.1)
##  yaml            2.1.19   2018-05-01 CRAN (R 3.4.4)
```