Are Nuclear Insertions of Mitochondrial Origin Pseudogenes?

Marta Sanchez Delgado
Máster Universitario en Bioinformática y Bioestadística (UOC-UB)

Área 29: Antropología Biológica

Xavier Jordana Comin
David Merino Arranz

05/06/2018

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Are Nuclear Insertions of Mitochondrial Origin Pseudogenes?*
Nombre del autor:	*Marta Sanchez Delgado*
Nombre del consultor/a:	*Xavier Jordana Comin*
Nombre del PRA:	*David Merino Arranz*
Fecha de entrega (mm/aaaa):	06/2018
Titulación::	*Máster Universitario en Bioinformática y Bioestadística (UOC-UB)*
Área del Trabajo Final:	*M0.128 TFM-Estadística y Bioinformática 29*
Idioma del trabajo:	Inglés
Palabras clave	*Human NUMTs; pseudogenes; Gene ontology and expression study*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

Las secuencias nucleares procedentes de DNA mitochondrial (NUMTs), son el resultado de una transferencia continua de fragmentos de DNA mitocondrial al núcleo. Durante estos años, se han realizado principalmente estudios filogenéticos de estas regiones pero, ¿Hay genes expresándose en estas regiones? En 2011, en un estudio liderado por C. Santos [11], se identificaron 755 NUMTs en el genoma humano. El principal objetivo de este trabajo final de máster ha sido la creación in-silico de una base de datos que contiene la lista de genes anotados dentro de estas 755 secuencias de NUMTs y su correspondiente perfil de expresión. Con este trabajo pretendemos mejorar nuestro conocimiento sobre el impacto de los NUMTs en el DNA nuclear mediante su potencial contribución en su transcriptoma.

La información de la base de datos se ha obtenido de las anotaciones en ENSEMBL y Gene Ontology (GO) mediante la combinación de Scripts de R personalizados y el uso del paquete BioMart (del repositorio Bioconductor). Por otro lado, los datos de expresión provienen del portal GTEx (RNA-seq). La mayoría de estos genes se encuentran anotados como pseudogenes en ENSEMBL, pero con este trabajo mostramos que 65 de estos genes se expresan al menos, en un tejido humano.

En conclusión, nuestros resultados muestran que no todos los genes codificados en los NUMTs son pseudogenes y que se requiere un cambio manual de, al menos, 63 genes clasificados como “pseudogenes” en *Ensembl-Biotype* puesto que son genes activos.

Abstract (in English, 250 words or less):

Nuclear mitochondrial DNA sequences (NUMTs) are the result of a continuous DNA transfer from mitochondria to the nucleus. Over the years, lots of studies had performed in NUMTs sequences at the DNA level but, are these regions encoding genes? In 2011, in a published work led by C. Santos [11], it was identified 755 NUMTs in the human genome. The main objective of the present master's final project has been the in-silico creation of a database containing the list of annotated genes within these 755 NUMTs sequences and their expression profile. With this work, we want to improve our knowledge about NUMTs impact on nuclear DNA by its potential contribution to the transcriptome.

The database information was extracted first from the ENSEMBL and Gene Ontology (GO) annotations by a combination of in-house R scripts and the use of BioMarts package (from Bioconductor project package repository). Additionally, by using the public information of RNA-seq from GTEx Portal database and in-house R script, we show the expression profile of our list of genes. Most of these genes are currently annotated as pseudogenes in ENSEMBL, but we show that a total of 65 genes are expressed in at least, one human tissue.

In conclusion, our results show that not all genes within NUMTs are pseudogenes and it is needed the manual change of at least, 63 genes classified as “pseudogene” in the Ensembl Biotype since these genes are expressed.

Índice

List of Figures **6**

List of Tables **7**

1. Introduction **8**
1.1. Context and justification of the Work 8
1.1.1. General description 8
1.1.2. Justification of the Project 8
1.2. Objectives of the Work 9
1.2.1. General objectives 9
1.2.2. Specific objectives 9
1.3. Approach and method followed 10
1.4. Working plan 11
1.4.1. Tasks 11
1.4.2. Calendar 12
1.4.3. Deadlines 13
1.4.4. Risk analysis 13
1.5. A brief summary of products obtained 14
1.6. A brief description of the additional chapter in the memory 16
2. Bioinformatics, genome browser and dataset resources used **18**
2.1. Bioconductor 18
2.2. Ensembl genome browser and the BioMart project 18
2.3. Gene Ontology Consortium 19
2.4. Genotype-Tissue Expression Portal (GTEx) 20
2.5. The initial data and the mitochondrial DNA 20
3. Results **23**
3.1. General workflow and data filtering 23
3.1.1. Filtering our annotated genes 23
3.1.2. Original mitochondrial genes 24
3.2. General descriptive analysis of our datasets 24
3.3. Gene ontology analysis 27
3.4. Expression profiling 27
3.4.1. Genes within NUMTs 27
3.4.2. Mitochondrial Genes 32
4. Discussion **34**
5. Conclusiones **36**
5.1. The general conclusion of the technical work 36
5.2. The general conclusion of the scientific results 36
5.3. Achieved objectives 36
5.4. Follow-up fo the working plan 37
5.5. Future work 37
6. Glossary **38**
7. Bibliography **39**
8. Supplementary information **41**

List of Figures

Figure 1. Gantt diagram 12

Figure 2. Mitochondrial origin of our NUMTs 21

Figure 3. Circular human mitochondrial DNA 22

Figure 4. Example of filtering genes within NUMTs 23

Figure 5. Ensembl-Biotype classification 25

Figure 6. Heat map with all expressed genes 28

Figure 7: Medium and highly expressed genes 29

Figure 8. Ensembl-Biotype classification of expressed genes 31

Figure 9. Heat map with the expression profile of all mitochondrial genes 33

Figure 10. Visualising NUMT_01.001 in UCSC genome browser 34

List of Tables

Table 1. Tasks and deadlines 11

Table 2. Genes annotated by chromosome 25

Table 3. NUMTs classified by gene content 25

Table 4. Genes annotated by NUMT 25

Table 5. List of genes in Biotype not classified as pseudogenes 26

Table 6. GO term and definition associated with some MTRNR2L proteins 27

Table 7. The expression level of the 452 genes included in GTEx 27

Table 8. Genetic features of medium and high expressed genes 30

Table 9. Genetic features of four low expressed genes 32

1. Introduction

1.1. Context and justification of the Work

1.1.1. General description

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 [1], [2]. These sequences vary in number and size through eukaryotic species, being a perfect tool for phylogenetic studies. Since mitochondrial and nuclear DNA have different mutation ratios, by studying the differences between NUMTs and mitochondrial DNA (mtDNA), evolutionary researchers reconstructed the approximate evolutionary moment when these sequences were inserted into the nucleus [3], [4]. Additionally, the polymorphisms of NUMTs are also a commonly used tool in human population genetic studies [5]. It is also important for the scientific community to be able to identify these regions in the reference genome browsers to avoid cross-contamination of mtDNA in nuclear DNA studies and vice versa [6]–[11].

Although these regions were reported for the first time more than thirty years ago, little is known about their insertion mechanism and its impact on the nuclear DNA [12]. Over the years, several studies were performed in NUMTs sequences at the DNA level but, are these regions encoding genes? More exhaustive studies are necessary to determine if there are genes within these regions and the impact on the genomic context caused by NUMTs insertions.

1.1.2. Justification of the Project

In 2011, in a published work led C. Santos [11], 755 NUMTs were identified in the human genome using BLAST. The authors compared the human mtDNA (NC_012920) against the human genome (GRCh37/hg19 assembly) and they described different aspects of this comparisons like frequency, distribution and size of NUMTs for each chromosome or % 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. As a result of this study, we want to generate a dataset with all relevant genetic content to be able to answer this question.

1.2. Objectives of the Work

1.2.1. General objectives

I. Apply the knowledge acquired during the master's degree in the big data management and the correct use of R bioinformatic repositories

II. Elucidate if NUMTs are originating pseudogenes

1.2.2. Specific objectives

I. To manage and study the initial information into its genetic context and annotated information:

   A. Explore and get familiar with initial data used in this work (from Ramos *et al.* [11])

   B. Define genetic context to be considered in our analysis

   C. Evaluate and select the online databases (UCSC genome browser tracks) relevant for the genetic study (selecting the most informative ones)

   D. Explore and evaluate Bioconductor and CRAN-R-project packages repositories for UCSC genome browser Tracks management

   E. Create R scripts to generate our data tables

   F. Perform a descriptive statistical analysis of our dataset

   G. Determine the main biological functions of NUMTs pseudogenes (Gene Ontology study)

II. To directly elucidate if genes within NUMTs are originating expressed genes and not only pseudogenes:

   A. Determine the main target tissues were NUMTs pseudogenes are expressing (GTEx)

   B. Classify and plot all data information to elucidate the degree of expression in the different tissues

   C. Compare expression profile of NUMTs genes with mitochondrial genes

1.3. Approach and method followed

Different strategies could be addressed to elucidate if NUMTs contain active genes. First, to generate the list of annotated genes within these sequences, different gene databases could be used (e.g. RefSeq genes, UCSC genes, Ensembl genes...). We have decided to generate a set of manageable data tables to facilitate consultation by future researchers unfamiliar with bioinformatics.

For the same reason, the R Script created to generate the dataset will be easily adaptable to obtain a new list of annotated genes within the new coordinates (Human Assembly GRCh37/hg19). For this propose, Bioconductor packages repository has been the main resources to manage and download information from BioMart project, including Ensembl ID and Gene Ontology annotations. Our election of Ensembl gene database is based on its continuous updating and its connection with GTEx, which contains the gene expression information in 53 tissues from RNA-seq of 8555 samples. The dataset generated at the end of this project comprise additional relevant information from genes containing in the corresponding mitochondria donor regions; location of NUMTs (intergenic, partial genes, introns); and GTEx information for expression means in each tissue.

All these information will be relevant to elucidate the potential role of the identified expressed genes in the case of having expression data from GTEx.

1.4. Working plan

1.4.1. Tasks

Table 1. Tasks and deadlines. List of tasks performed for this Final Master Project and its corresponding deadlines.

FIRST DEFINITIONS	**PEC0**
Selecting the topic, the problem to solve and aims	21/02/2018 - 05/03/2018
WORKPLAN	**PEC1**
Defining genetic context to be considered	06/03/2018 - 19/03/2018
WORK DEVELOPMENT	
Phase I: Dataset generation	**PEC2**
Exploring online databases (UCSC genome browser Tracks/Ensembl)	20/03/2018 - 26/03/2018
Dataset creation by R Scripts (**BioMart package - Bioconductor - Including GO terms**)	27/03/2018 - 16/04/2018
Preliminary descriptive analysis	16/04/2018 - 23/04/2018
Phase II: Dataset analysis	**PEC3**
Filtering gene list (genes within NUMTs)	23/04/2018 - 08/05/2018
Expression study (GTEx Gene repository)	16/04/2018 - 14/05/2018
Statistical analysis of gene expression (GTEx) - graphical representation	14/05/2018 - 21/05/2018
MEMORY	**PEC4**
Writing the memory	22/05/2018 - 06/06/2018
PRESENTATION	**PEC5**
Elaborating the presentation	07/06/2018 - 13/06/2018
Public defence	14/06/2018 - 25/06/2018

1.4.2. Calendar

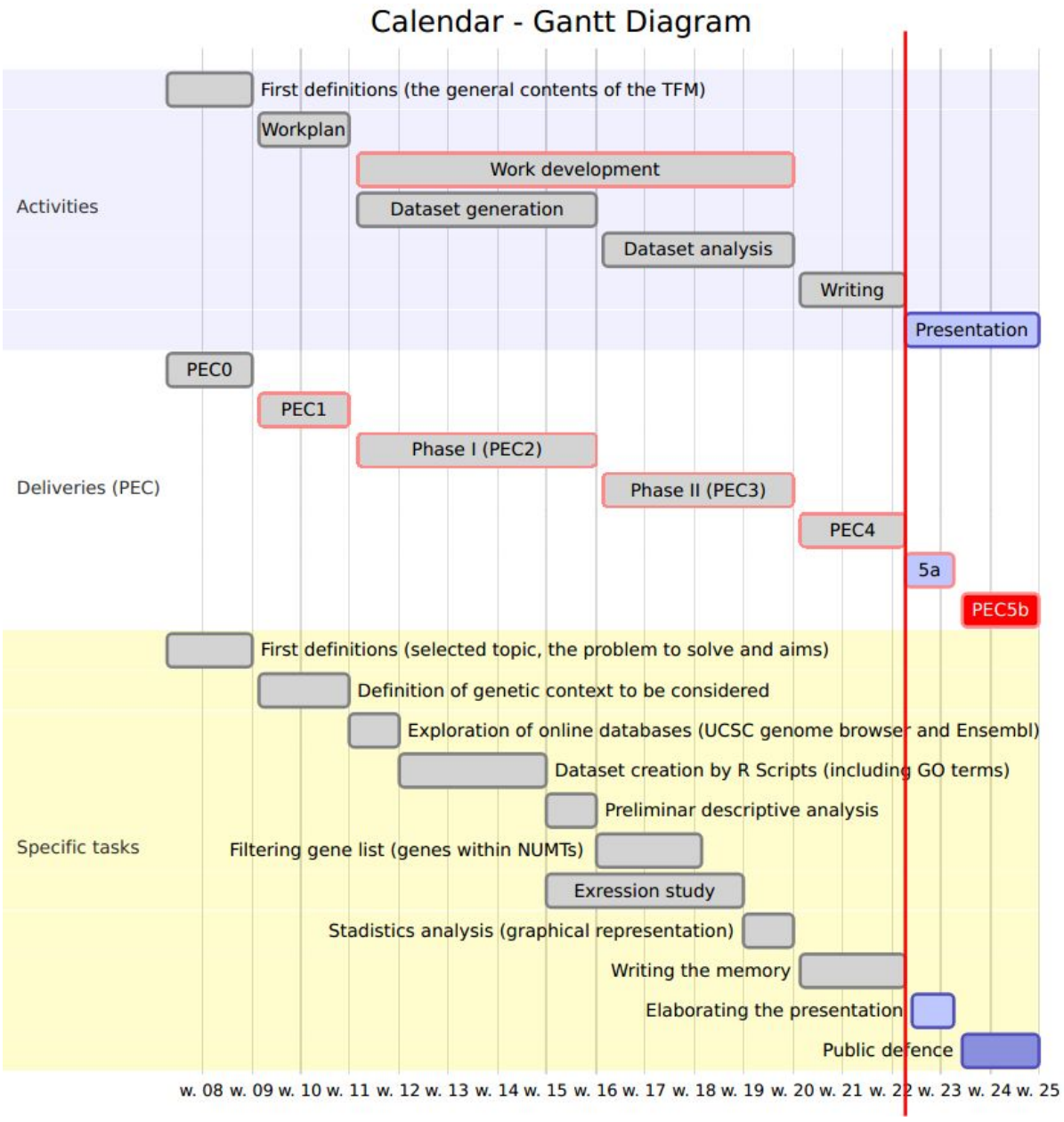

Figure 1. Gantt diagram. Temporal planning for all tasks performed in this final project. Each task is represented in a colour filled box: grey if it is already achieved, light blue if it is active, dark blue for future tasks and red if it is critical. The diagram is divided into three sections. The first section contains the general activities (blue shaded section), the second section includes dates of each delivery (white section with red boxes for each PEC), and the final section includes specific tasks carried out throughout the semester (orange shaded section). At the bottom, it is indicated the week year numbering. Grey vertical lines indicate the Monday of each week. The first line is 26/02/2018 (week 8). **Updated: 05/06/2018**

1.4.3. Deadlines

PEC0. First definitions: *selected topic, the problem to solve and aims*
(21/02/2018 - 05/03/2018)

PEC1. Workplan: *definition of genetic context to be considered*
(06/03/2018 - 19/03/2018)

PEC2. Work development (Phase I): *dataset generation*
(20/03/2018 - 23/04/2018)

PEC3. Work development (Phase II): *dataset analysis*
(24/04/2018 - 21/05/2018)

PEC4. Writing the memory
(22/05/2018 - 06/06/2018)

PEC5a. Elaborating the presentation
(07/06/2018 - 13/06/2018)

PEC5b. Public defense
(14/06/2018 - 25/06/2018)

1.4.4. Risk analysis

I. Inability to generate a script to generate any of the data tables. *In this case, we will try to find alternative databases*

II. Negative results: NUMTs do not contain genes or pseudogenes. *If we do not find genes in these regions (or we find very few genes) this project should change completely. Instead of mainly focus our study on to the expression analysis, we will focus it on the conservation study, the presence of repetitive sequences and epigenetic analysis.*

1.5. A brief summary of products obtained

At the end of this project you will get the following products:

I. NUMTs-ID-bed.txt

This file contains the coordinates of all NUMTs including its localisation in mitochondrial genome adapted to bed format to its uploaded in UCSC genome browser. Once you upload this file in the UCSC genome browser, you will visualise it in a new track on your screen together with other genomic data from all tacks activated. This document is available in GitHub platform:

https://github.com/msanchezd/UOC-Bioinformatics/tree/TFM-documents

II. DynamicReport.Rmd (.html)

This document (also in the supplementary section of this memory) contains the description of all data tables and its content, together with some figures describing Ensembl Biotype, GO Terms and expression profile (GTEx) from all genes within the given initial coordinates. Following its instructions, by changing the name of the input document and download this .Rmd file together with .bib and your new .cvs containing the coordinates, it is easy to generate a new .html dynamic report and all data-tables with the new input information. This document is available in GitHub platform:

https://github.com/msanchezd/UOC-Bioinformatics/tree/Dynamic-Report

III. Scripts.R

Document with all scripts designed for these project (all the code in this document is the scrips in the last part of the Supplementary section). The first part contains all scripts to generate the tables and data in DynamicReport document and the second part is adapted specially for our data, including all figures and additional data filtering. This document is available in GitHub platform:

https://github.com/msanchezd/UOC-Bioinformatics/tree/TFM-documents

IV. Data tables: A total of 16 data tables containing the output of analysis. All of them also available in GitHub platform:

https://github.com/msanchezd/UOC-Bioinformatics/tree/TFM-documents

Tables obtained from the Dynamic Report (*dimensions in Row x Col*)

File 1 - All_attributes.txt (1416 x 3): list of attributes included in BioMarts Bioconductor package.

File 2 - All_filters.txt (303 x 2): list of filters included in BioMarts Bioconductor package.

File 3 - gene_results.txt (1155 x 9): table of all genes identified before our filtering. The table includes id NUMT, chromosome coordinates for each gene, strand (+ or -), HGNC symbol, Ensembl Gene ID with and without version and transcript count.

File 4 - up_gene_results.txt (891 x 9): table of all genes identified upstream genes in "gene_results.txt" (between 100-1000 bp upstream). The table includes id NUMT, chromosome coordinates for each gene, strand (+ or -), HGNC symbol, Ensembl Gene ID with and without version and transcript count.

File 5 - down_gene_results.txt (905 x 9): table of all genes identified downstream genes in "gene_results.txt" (between 100-1000 bp downstream). The table includes id NUMT, chromosome coordinates for each gene, strand (+ or -), HGNC symbol, Ensembl Gene ID with and without version and transcript count.

File 6 - genes.txt (456 x 1): Ensembl Gene ID of all genes within our NUMTs after filtering.

File 7 - go_results.txt (22 x 5): GO id and descriptions for our genes.

File 8 - phenotype_results.txt (456 x 5): table of all genes within NUMTs (after filtering) including HGNC symbol, Ensembl Gene ID version, Transcript count, Gene Biotype, Gene description (from different sources).

File 9 - mean_tpm_GTEx.txt (452 x 58): genes within NUMTs (after filtering) that have expression information in GTEx portal. The table includes the expression in Transcripts per Million (TPM) of all 53 tissues, a mean per gene and the sum of all TPM per gene including all tissues.

File 10 - subset_expressed.txt (72 x 58): genes within NUMTs (after filtering) that are expressed in at least one tissue (>= 0.5 TPM). The table includes the expression in TPM of all 53 tissues, a mean per gene and the sum of all TPM per gene including all tissues.

File 11 - FINAL_OUTPUT_TABLE.txt (998 x 75): the final table including the 756 genes, its coordinates, its expression data, different nomenclatures, Ensembl-Biotype and GO term.

<u>Additional tables especifically obtained for this master thesis</u>

The following tables are obtained from the second part of the Script.R (showed in supplementary section):

File 12 - gene_result_mt.txt (3954 x 9): table of all mitochondrial genes overlapping with original mitochondrial NUMT sequences. Since generally, most NUMTs are originated by several mitochondrial genes, the table have 3954 rows (and the mitochondria have 38 annotated genes). The table includes id NUMT, chromosome coordinates for each gene, strand (+ or -), HGNC symbol, Ensembl Gene ID with and without version, transcript_count and Ensembl biotype classification.

File 13 - total_overlaping.txt (1007 x 11): all nuclear and mitochondrial genes which partially overlaps (without any cutoff). The table contains HGNC symbol for nuclear and mitochondrial genes and adapted coordinates for nuclear and mitochondrial genes. The start and end position of coordinates are: the first four (NUMTs in chromosomes 1 to 9) or five numbers (NUMTs

in chromosomes 10 to 22, 23 for X and 24 for Y) indicates the id NUMT (11001XXXXX) the last five numbers corresponds to the position of the gene within the NUMT (XXXXX00352). The length of each gene and the full overlapping region is indicating, together with % of overlap respect the mitochondrial or nuclear gene.

File 14 - all_data_70.txt (1007 x 11): combination of table "FINAL_OUTPUT_TABLE.txt" with all new data from comparing nuclear genes and mitochondrial genes from "total_overlaping" but after filtering. We only include overlapping genes if 70% of the nuclear gene is formed by the corresponding mitochondrial gene or the cases that at least the 70% of a mitochondrial gene it is present in the nuclear gene.

File 15 - mito_genes.txt (37 x 1): Ensembl Gene ID of all mitochondrial genes annotated.

File 16 - mean_tpm_GTExMITO.txt (37 x 58): Mitochondrial genes with expression data in GTEx portal. The table includes the expression in TPM of all 53 tissues, a mean per gene and the sum of all TPM per gene including all tissues.

1.6. A brief description of the additional chapter in the memory

★ **Bioinformatics, genome browser and dataset resources used**

General description of all online and bioinformatic resources used during this work, their potential and accessibility and the origin of the initial dataset (original study [11]):

I. Bioconductor
II. Ensembl genome browser and the BioMart project
III. Gene Ontology Consortium
IV. Genotype-Tissue Expression Portal
V. The initial dataset and the mitochondrial DNA

★ **Results**

I. General workflow and data filtering

Description of the steps followed for filtering our data by in-house R-scripts to generate the final list of genes within the NUMTs and determine the mitochondrial gene contribution to them.

II. General descriptive analysis of our datasets

Before to determine if the genes are expressed or not, in this subsection it is shown an initial analysis of general characteristics (e.g. number of genes annotated per chromosome and NUMT or the classification of these genes in Ensembl-Biotype).

III. Gene ontology analysis

Once we got a list of genes, we generated a list of GO Terms and determine if there are biological functions overrepresented in potential genes originated by NUMTs.

IV. Expression profiling

The key analysis performed in this study was to determine if any of these annotated genes within NUMTs is expressed in at least, one tissue. In this subjection, the annotated genes were classified in low, middle and high expressed. Additionally, it was also shown expression profile of all mitochondrial genes in the different human tissues.

★ Discussion

Once we finish all our expression analysis, we discussed the biological and technical implications of our results.

★ Conclusions

List of main conclusions we can draw from the present work.

2. Bioinformatics, genome browser and dataset resources used

2.1. Bioconductor

The Bioconductor project started in 2001 and it was mainly created to provide a supervised tool for the analysis and comprehension of high-throughput genomic data [13]. Bioconductor uses the R statistical programming language and its content is updated twice each year. It is an open source and open development software project and is overseen by a core team at Roswell Park Cancer Institute and by other members of US and other international institutes [14].

The last updated version of Bioconductor contains 1560 software packages and the goals of this project are related to the power of R statistical programming:

- To provide access to powerful statistical and graphical methods for the analysis of genomic data.
- To facilitate the inclusion of biological metadata (annotated in public genome browsers or databases) in the analysis of genomic data.
- To provide a common software platform.
- To further scientific understanding by producing high-quality documentation and reproducible research.
- To train researchers on computational and statistical methods for the analysis of genomic data.

Additionally, Bioconductor has an active user community continuously interacting with its content and anyone could become a developer, by contributing to packages resource or associated documentation. All these applications will be supervised by Bioconductor experts. But Bioconductor also facilitates the interaction between scientific community by organizing every year a conference to show the current developments within and beyond Bioconductor, and by linking together different groups with common goals to facilitate collaborations.

2.2. Ensembl genome browser and the BioMart project

The Ensembl genome database project is the result of a collaborative work between the European Bioinformatic Institute and the Wellcome Trust Sanger Institute (EMBL-EBI) started in 1999. Nowadays, Ensembl is a genome browser for vertebrate genomes and its annotate genes are continuously updated and linked with other genomic tools like Bioconductor or the GTEx database. The current genome assembly is GRCh38, however, we use the previous assembly since it is the most complete:

GRCh37 Full Feb 2014 archive with BLAST, VEP and BioMart

The GRCh37 assembly contains 20,805 coding genes, 196,501 gene transcripts, 48,597 Genscan gene predictions, 328,852,510 short variants and 5,806,176 structural variants. Additionally, the Ensembl automatic annotation system classifies genes and transcripts into biotypes including protein-coding, pseudogene, processed pseudogene, micro RNA

(miRNA), ribosomal RNA (rRNA), small-nuclear RNA (snRNA) and small-nucleolar RNA (snoRNA). In the GRCh37 assembly, 14,181 annotated genes are classified as pseudogenes. Although in some cases, for human, mouse, rat and pig, Ensembl incorporate manual annotation from Havana, in most of the cases it is in-silico automatic classification.

Finally, it is important to highlight that Ensembl has an online tool with a user-friendly interface: the BioMart tool. In this webpage you can select:

- **Dataset:** Specific genome assembly
- **Filters:** Here you will introduce your ***input data*** and indicate how you want to filter your output (from a list of gene IDs, chromosomes…)
- **Attributes:** Here you will select the information you want to extract from your input data. Attributes include data from Ensembl and external data from other databases like Gene Ontology Consortium, EntrezGene ID, HUGO Gene Nomenclature Committee ID (HGNC) or RefSeq.

This Ensembl took is based on the BioMart project, which is a community-driven project to provide unified access to distributed research by providing free software and data services to the international scientific community, facilitating scientific collaborations and helping the discovery process. But BioMart is not only integrated into Ensembl as a tool, it is also integrated into Galaxy, Cytoscape, Taverna and also in Bioconductor, as a complete R package. In this Master's final project, the BioMart Bioconductor package has been used indicating **GRCh37 assembly version** and **Homo sapiens Ensembl genes (version 92).** Different ***filters*** **and** ***attributes*** were chosen depending on what we were downloading.

2.3. Gene Ontology Consortium

The Gene Ontology (GO) project provides a comprehensive resource currently available for computable knowledge regarding the gene functions and gene products.

When you provide a large list of genes included in this database, you can perform an enrichment analysis powered by PANTHER. This analysis consists in the classification of the genes provided depending on its functional classification (given by GO) and comparing to the characteristics of all genes included in GO, you will know if there is a significant representation of a specific biological process, molecular function or cellular component (depending the analysis you chose) in your list of genes.

As we will see in the Results section, there is only a single family of genes within our NUMTs included in GO, which makes the PANTHER statistical analysis impossible.

2.4. Genotype-Tissue Expression Portal (GTEx)

The portal for the Genotype-Tissue Expression (GTEx) project provides free access to gene expression and quantitative trait loci from 53 human tissues. This project was supported by the Common Fund of the Office of the Director of the National Institutes of Health, and by NCI, NHGRI, NHLBI, NIDA, NIMH, and NINDS.

The GTEx Project was founded in 2015 aiming to provide to the scientific community a resource with which to study human gene expression and regulation and its relationship to genetic variation. This project is continuously collecting and analysing multiple human tissues from donors who are also genotyped, to assess genetic variation within their genomes. By analysing global RNA expression within individual tissues and treating the expression levels of genes as quantitative traits, variations in gene expression that are highly correlated with genetic variation can be identified as expression quantitative trait loci or eQTLs. To achieve its aims, the GTEx Consortium is continuously incorporating new working groups, facilitating scientific collaborations and increasing the available datasets. The current data provided by GTEx Portal is associated with coordinates and IDs fromhg19/GRCh37 human genome reference and the 1000 genomes project.

The data used for the analyses described in this Master's project were obtained from ***Datasets*** → ***download*** section in GTEx project. Specifically, the file containing the median TPM by tissue (GTEx Analysis V7) in ***RNA-Seq Data*** subsection. The access to GTEx files is free, but you need to sign in. The name of the document used is:

GTEx_Analysis_2016-01-15_v7_RNASeQCv1.1.8_gene_median_tpm.gct.gz

In the ***Datasets*** → ***download*** section from GTEx Portal you can also download other expression data (e.g. gene read counts or transcripts TPM) to locally work on your computer. However, to fully access to this GTEx subsection, the user needs to sing in, which normally is automatically identified if you already sign in your web browser.

2.5. The initial data and the mitochondrial DNA

In the initial work published in 2011 by Ramos et al [11], it was performed a BLAST by comparing The Human mtDNA Reference Sequence (NC_012920) against the human RefSeq Genomic database at NCBI (GRCh37). It was detected NUMTs in all human chromosomes but with different freqüency. For example, chromosome 2 is where more NUMTs insertions were described, but the size of these insertions was lower than the mean.

As we can see in Figure 2 showing the corresponding mitochondrial regions that originate the 755 NUMTs described in the initial study, the insertions were generated from all the extension of mtDNA, being some specific regions highly represented.

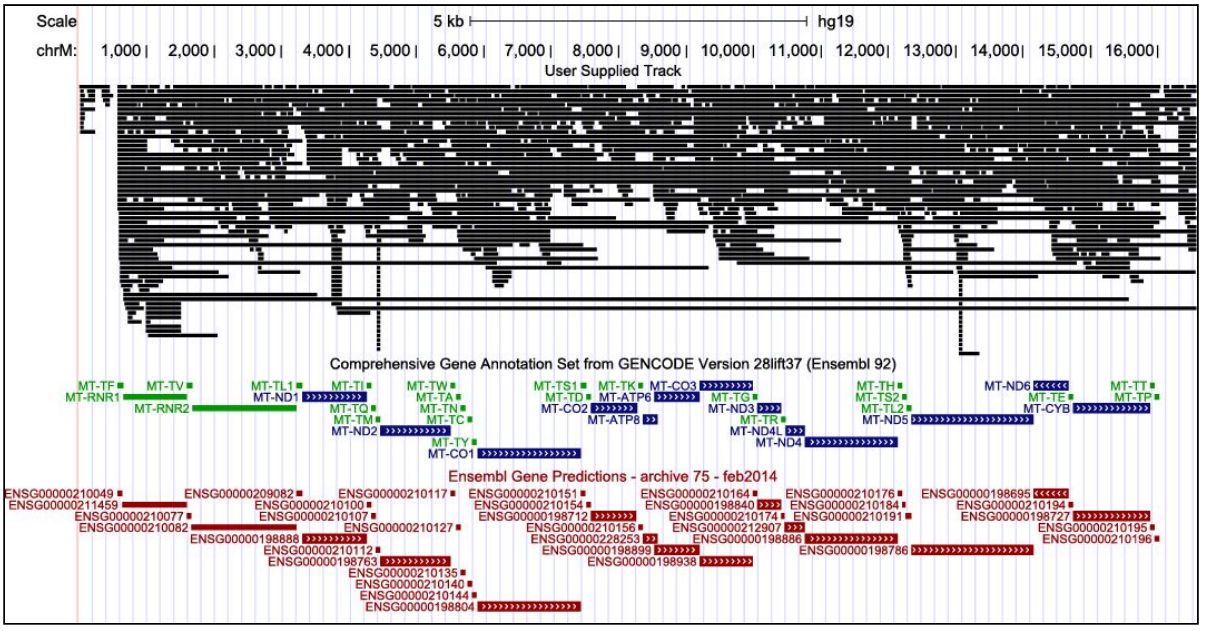

Figure 2. Mitochondrial origin of our NUMTs. Screenshot of our custom track (UCSC Genome browser) containing all NUMTs coordinates and its corresponding mitochondrial origin regions. The full extent of the mitochondrial DNA is represented.

This NUMTs identified are the most recent insertions from the mtDNA and some of them could be human or primate-specific. Thought evolution, the eukaryotic DNA has been receiving different mtDNA insertions, in some cases, the mitochondria have lost the corresponding DNA sequence and nowadays, some mitochondrial origin genes are exclusively in the eukaryotic nucleus [15].

The modern human mtDNA contain genes encoding for 13 proteins (Figure 3), which are structural subunits of the mitochondrial respiratory chain (to produce ATP). Additionally, since the genetic code of the mitochondria is different from the genetic code of the nuclear genome, the mitochondrial DNA includes corresponding transfer RNAs (tRNAs). The 22 tRNA and 2 rRNA genes are located between the peptide-encoding genes [16]. However, our mitochondria contain about 1000 proteins, most of them are nuclear-encoded which are translated on cytosolic ribosomes and actively imported and sorted into mitochondrial subcompartments [17], [18].

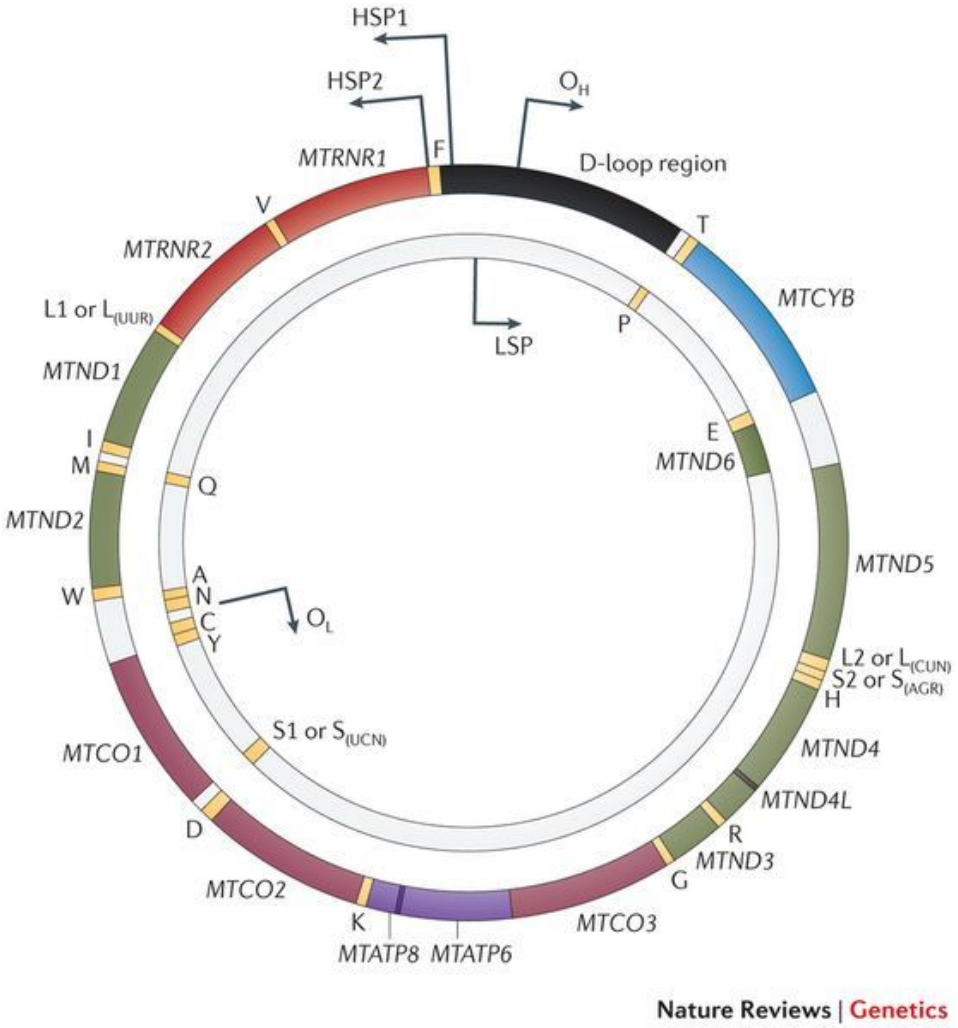

Figure 3. Circular human mitochondrial DNA. Representation of all genes annotated in the human mitochondrial DNA and the initiation sites for both heavy and light strands transcription [16].

3. Results

3.1. General workflow and data filtering

3.1.1. Filtering our annotated genes

By using BioMart package (from Bioconductor), all the genes annotated in Ensembl which coordinates overlap with our initial coordinates are selected. However, since this list also includes large gene coding proteins where the NUMTs are probably located in intronic regions, we decide to perform a first filtering to obtain an accurate list of genes (Scripts in PART 1 of the supplementary data). For this propose, 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, we eliminate from the initial list those genes also present upstream and downstream the initial coordinates. In figure 4, *SPTLC1* (pink names) is the only one present in all lists, so is the only one deleted. In the case of *MTC03P29* (pink names), its coding region is formed by two different NUMTs, as a consequence, this gene will be in the upstream least of NUMT_09.025 and the downstream list of NUMT_09.026. However, we conserve this gene because we consider the possibility of having a gene formed by more than one NUMTs and we delete exclusively "paired annotations," i.e., genes present in upstream and downstream part of a specific NUMTs. With this approach, if we have a gene originated from more than three different NUMTs, we will not lose this gene because it is not in one of the downstream or upstream lists for the initial and final NUMT.

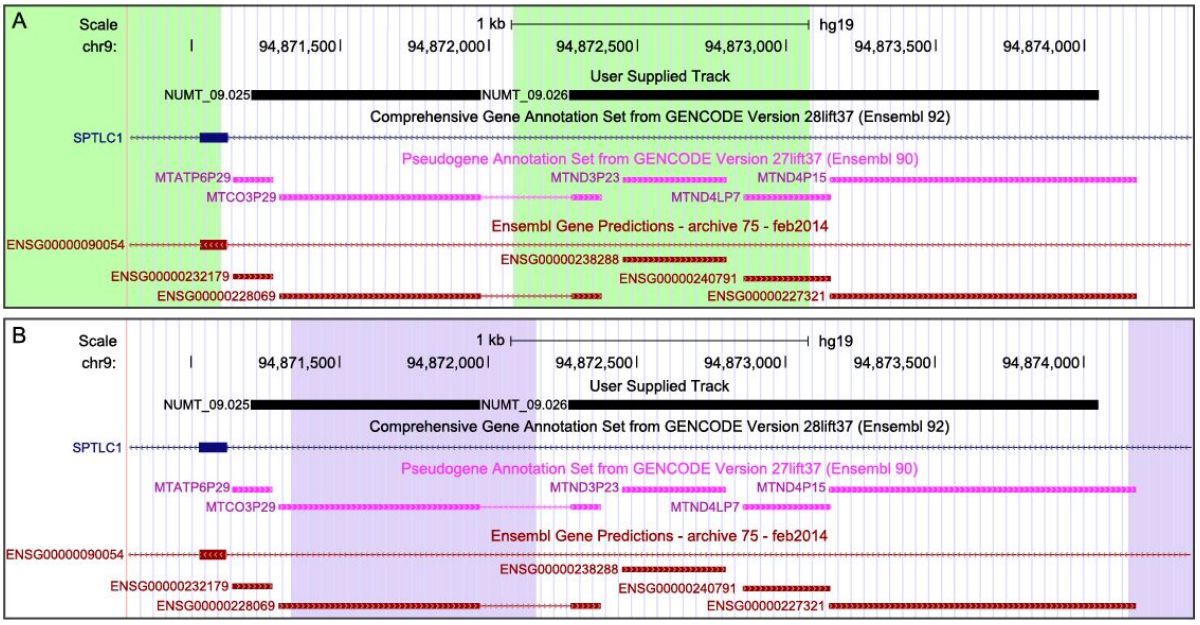

Figure 4. Example of filtering genes within NUMTs. Screenshot of our custom track (UCSC Genome browser) containing all NUMTs coordinates and its corresponding mitochondrial origin regions. In the example, our first approach will recognise a total of 6 Ensembl genes (in dark red). With our filtering, we eliminate the big gene where our NUMTs are overlapping with the intronic region and keep the other 5 genes. **(A)** In green,

upstream and downstream region for NUMT_09.025 and **(B)** In purple, upstream and downstream region for NUMT_09.026.

3.1.2. Original mitochondrial genes

In parallel, the coordinates of the different mitochondrial genes were also extracted and associated with its corresponding NUMT. Using the information from NUMTs coordinates and the direction of the genes (strand + or -), a search for overlap in NUMTs genes and its corresponding mitochondrial genes were performed (Scrips in PART 2 of supplementary data):

- ➔ 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)

To accurate the analysis and to focus on nuclear genes that are mainly originated from mitochondrial genes, we filtered for regions where, the 70% of the nuclear gene is formed by the corresponding mitochondrial gene or the cases that at least the 70% of a mitochondrial gene it is present in the nuclear gene.

3.2. General descriptive analysis of our datasets

Before the filtering, in “gene_results.txt”, 733 different genes were detected in our analysis. After discriminating big genes included in our first list and between 100-1000 bp upstream and downstream each NUMT, we obtain a list of 456 genes differently distributed in the chromosomes, being the chromosome 2 the one with more genes (Table 2).

The genes annotated are differently distributed within NUMTs, being in total, 468 NUMTs which do not contain any gene (see Table 3). In general, a single gene is annotated for those NUMTs containing genes, but exceptionally, it is identified 1 NUMT with 8 annotated genes (NUMT_17.013), 1 NUMT with 10 (NUMT_02.043) annotated genes and 2 NUMTs with 11 annotated genes (NUMT_04.035 and NUMT_05.022).

Finally, we also identified some genes formed by more than one NUMT (Table 4). After eliminating 277 genes before the filtering, 401 genes are included in only 1 NUMT, 50 genes are originated from 2 different NUMTs, only 4 genes are included in three NUMTs (*MTND5P28, MTND5P1, RP3-433F14.2* and *Z95114.7*) and a single gene is originated by 4 different NUMTs (*MTND4P31*).

For our list of 456 genes within NUMTs, we additionally search for annotated classifications of gene features. Since our search was focussed on Ensembl annotations, we check the corresponding Biotype classification for our list of genes (Figure 5).

Table 2. Genes annotated by chromosome.

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

Table 3. NUMTs classified by gene content. The table indicates how many NUMTs contains a specific number of genes (starting with the number of NUMTs without genes and finishing with the number of NUMTs containing eleven genes).

Any gene	One gene	Two genes	Three genes	Four genes	Five genes	Six genes	Seven genes	Eight genes	Nine genes	Ten genes	Eleven genes
468	183	59	19	11	3	3	6	1	0	1	1

Table 4. Genes annotated by NUMT. The table indicates how many genes are annotated in one, two, three or four NUMT (any gene is formed by five or more NUMTs).

Genes in one NUMT	Genes in two NUMTs	Genes in three NUMTs	Genes in four NUMTs
401	50	4	1

As we can see in Figure 5, 436 genes (95.6% of our filtered list of genes) within NUMTs are classified as pseudogenes (they are similar to known proteins but contain a frameshift and/or stop codon(s) which disrupts the ORF) by Ensembl Biotype. These genes are probably initially annotated by in-silico approaches due to its similarity with mitochondrial genes and are classified as pseudogenes due to its different genomic code, but in the following section we compare this information with GTEx data to elucidate if all of them are pseudogenes or some of them are expressed (even though if they are translated with nucleus machinery, these genes will encode proteins with different structure than the corresponding mitochondrial proteins).

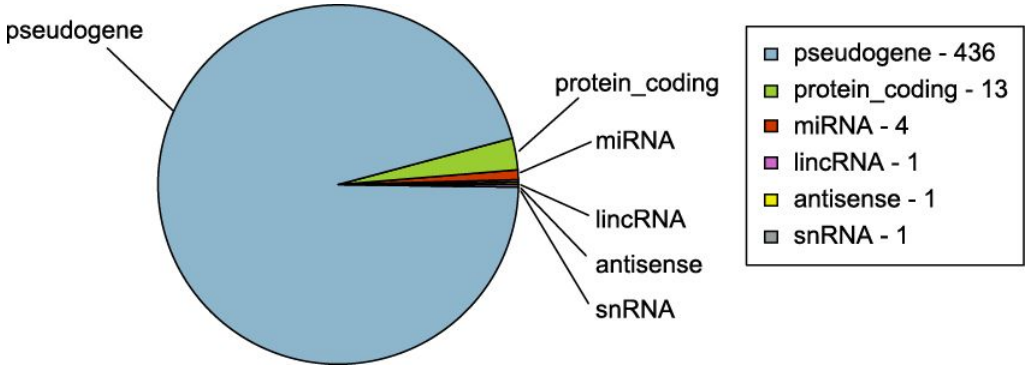

Figure 5. Ensembl-Biotype classification. Only 436 out of 456 genes identified are classified as pseudogenes in Ensembl-Biotype. For the other annotated genes, 13 corresponds to protein-coding genes, 4 miRNAs, 1 antisense, 1 lincRNA and 1 snRNA.

From the genes annotated in Biotype, the only genes not classified as pseudogenes are the Humanin-like genes (*MTRNR2L*), four miRNA, a small nuclear RNA (snRNA) and a large intergenic non-coding RNAs (lincRNAs). However, some of these annotated genes are specifically described as pseudogenes in HGNC (see Table 5).

Table 5. List of genes in Biotype not classified as pseudogenes.

Gene symbol (GTEx)	Ensembl ID	chr	Start	End	Strand	Gene biotype	description from HGNC
MTRNR2L1	ENSG00000256618.1	17	22022437	22023991	1	protein_coding	*MT-RNR2-like 1*
MTRNR2L2	ENSG00000271043.1	5	79945819	79946855	-1	protein_coding	*MT-RNR2-like 2*
MTRNR2L3	ENSG00000256222.1	20	55933496	55934878	-1	protein_coding	*MT-RNR2-like 3*
MTRNR2L4	ENSG00000232196.2	16	3421053	3422283	-1	protein_coding	*MT-RNR2-like 4*
MTRNR2L5	ENSG00000249860.2	10	57358750	57360488	1	protein_coding	*MT-RNR2-like 5*
MTRNR2L6	ENSG00000270672.1	7	142374104	142375550	1	protein_coding	*MT-RNR2-like 6*
MTRNR2L7	ENSG00000256892.1	10	37890366	37891859	-1	protein_coding	*MT-RNR2-like 7*
MTRNR2L8	ENSG00000255823.1	11	10529434	10530723	-1	protein_coding	*MT-RNR2-like 8*
MTRNR2L9	ENSG00000255633.3	6	62284008	62284534	1	protein_coding	*MT-RNR2-like 9 (pseudogene)*
MTRNR2L10	ENSG00000256045.1	X	55207824	55208944	-1	protein_coding	*MT-RNR2-like 10*
MTRNR2L11	ENSG00000270188.1	1	238107024	238108575	-1	protein_coding	*MT-RNR2-like 11 (pseudogene)*
MTRNR2L12	ENSG00000269028.2	3	96335981	96337000	-1	protein_coding	*MT-RNR2-like 12 (pseudogene)*
MTRNR2L13	ENSG00000270394.1	4	117220016	117221520	1	protein_coding	*MT-RNR2-like 13 (pseudogene)*
RP11-551L14.6	ENSG00000256984.1	12	31165437	31168601	1	lincRNA	
*MIR4461**	ENSG00000263963.1	5	134263729	134263802	1	miRNA	*microRNA 4461*
- no name -	ENSG00000264338.1	X	125606792	125606865	-1	miRNA	
- no name -	ENSG00000264839.1	5	99386069	99386142	1	miRNA	
MIR4484	ENSG00000265092.1	10	127508309	127508391	1	miRNA	*microRNA 4484*
RNU6-656P	ENSG00000252594.1	8	47742672	47742778	1	snRNA	*RNA, U6 small nuclear 656, pseudogene*
RP11-846F4.12	ENSG00000243655.2	13	22023889	22024656	-1	antisense	

**MIR4461 is annotated twice in HGNC, but the other corresponding ENSEMBL code, ENSG00000198868 (chr5:134263720-134264016, negative strand) is associated with MTND4LP30 in GeneCard database and classified as "pseudogene" in Ensembl Biotype.*

3.3. Gene ontology analysis

From all our list of genes within NUMTs sequences, only 11 are annotated in the Gene Ontology Consorcium. These genes are most of the *MTRNR2L* classified as "protein-coding" in Ensembl biotype (except *MTRNR2L11* and *MTRNR2L13*, that, in fact, as is highlighted in Table 5, are two of the ones considered as pseudogenes in HGNC). All these MTRNr2L proteins are associated with two GO terms: "Extracellular region" and "cytoplasms" (see Table 6). These means that all these proteins were detected in the extracellular space and cytoplasm. In the "localisation" section of GeneCards database, we can find the confidence of this localization, for example, for *MTRNR2L1*.

Table 6. GO term and definition associated with some MTRNR2L proteins

go_id	name_1006	definition_1006
GO:0005576	extracellular region	The space external to the outermost structure of a cell. For cells, without external protective or external encapsulating structures, this refers to space outside of the plasma membrane. This term covers the host cell environment outside an intracellular parasite. [GOC:go_curators]
GO:0005737	cytoplasm	All of the contents of a cell excluding the plasma membrane and nucleus, but including other subcellular structures. [ISBN:0198547684]

3.4. Expression profiling

3.4.1. Genes within NUMTs

By using the mean of expression in Transcript per million (TPM - normalization method for RNA-seq, which means that "for every 1,000,000 RNA molecules in the RNA-seq sample, x came from this gene/transcript."-) from GTEx Portal (from the file "GTEx_Analysis_2016-01-15_v7_RNASeQCv1.1.8_gene_median_tpm.txt.gct" in download GTEx Portal section), the cutoff to define a gene expressed was 0.5 TPM in at least one tissue (see Figure 6). This is the de default minimum expression level defined by the EMBL-EBI Expression Atlas.

Additionally, also following the EMBL-EBI Expression Atlas criteria, genes were classified in low expressed (between 0.5 and 10 TPM), medium expressed (>10 to 1000 TPM, see Figure 7B) and high expressed (more than 1000 TPM, see Figure 7A). In total, GTEx includes expression data for 452 of our genes and 72 are expressed (see Table 7).

Table 7. The expression level of the 452 genes included in GTEx. For the 456 genes annotated in Ensembl, 452 are included in RNA-seq data from GTEx and 72 of them are expressed in at least one tissue (> 0.5 TPM).

Highly expressed (>1000 TPM)	Medium expressed (>10 to 1000 TPM)	Low expressed (between 0.5 to 1000 TPM)	Not express genes
2	9	61	380

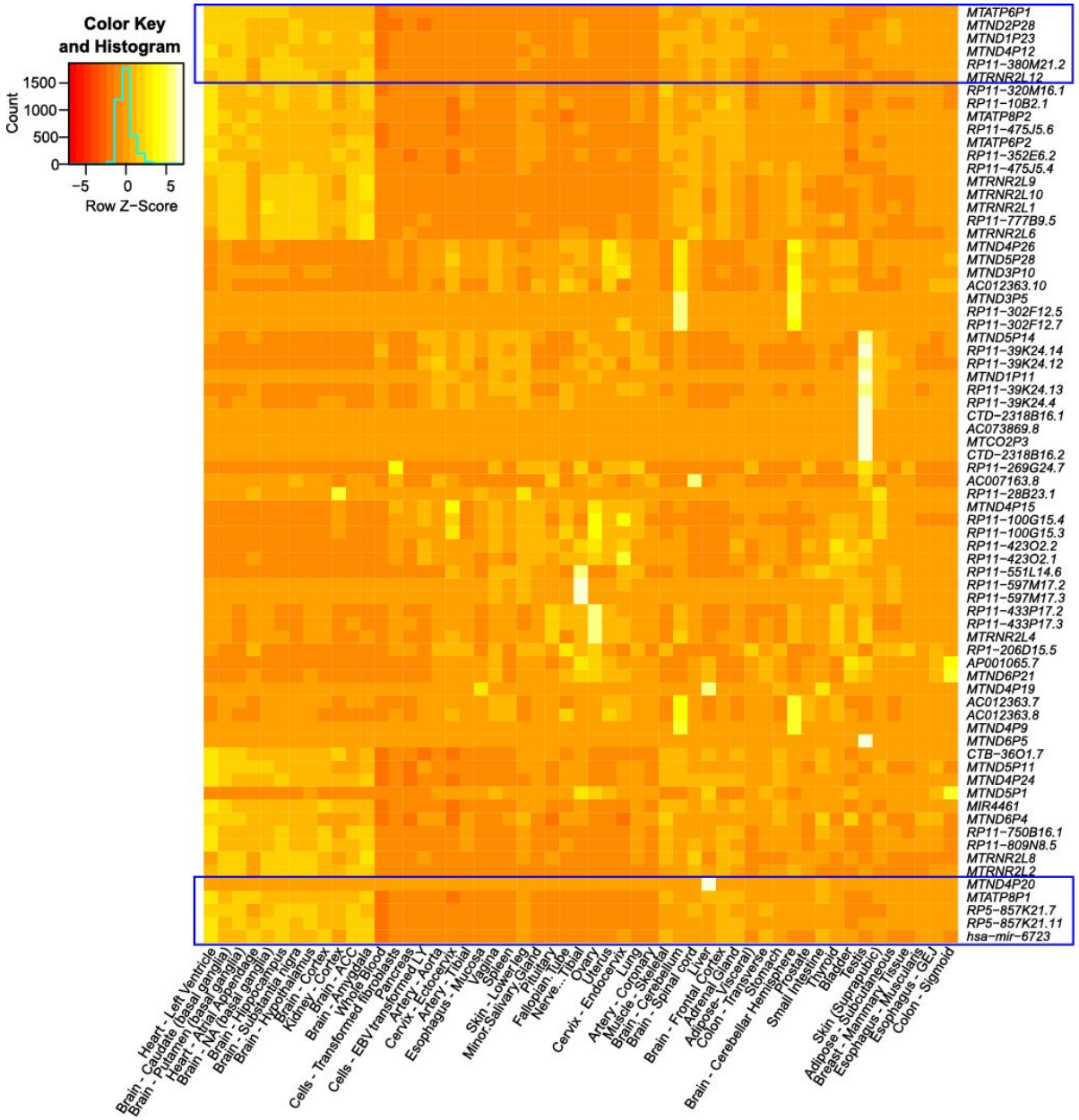

Figure 6. Heat map with all expressed genes (more than 0.5 TPM in at least 1 tissue). Z-score ($z = (x - \mu) / \sigma$) normalized by row, i.e. for each gene independently based on its expression profile in the different tissues. Blue boxes represents the highly and middle expressed genes.

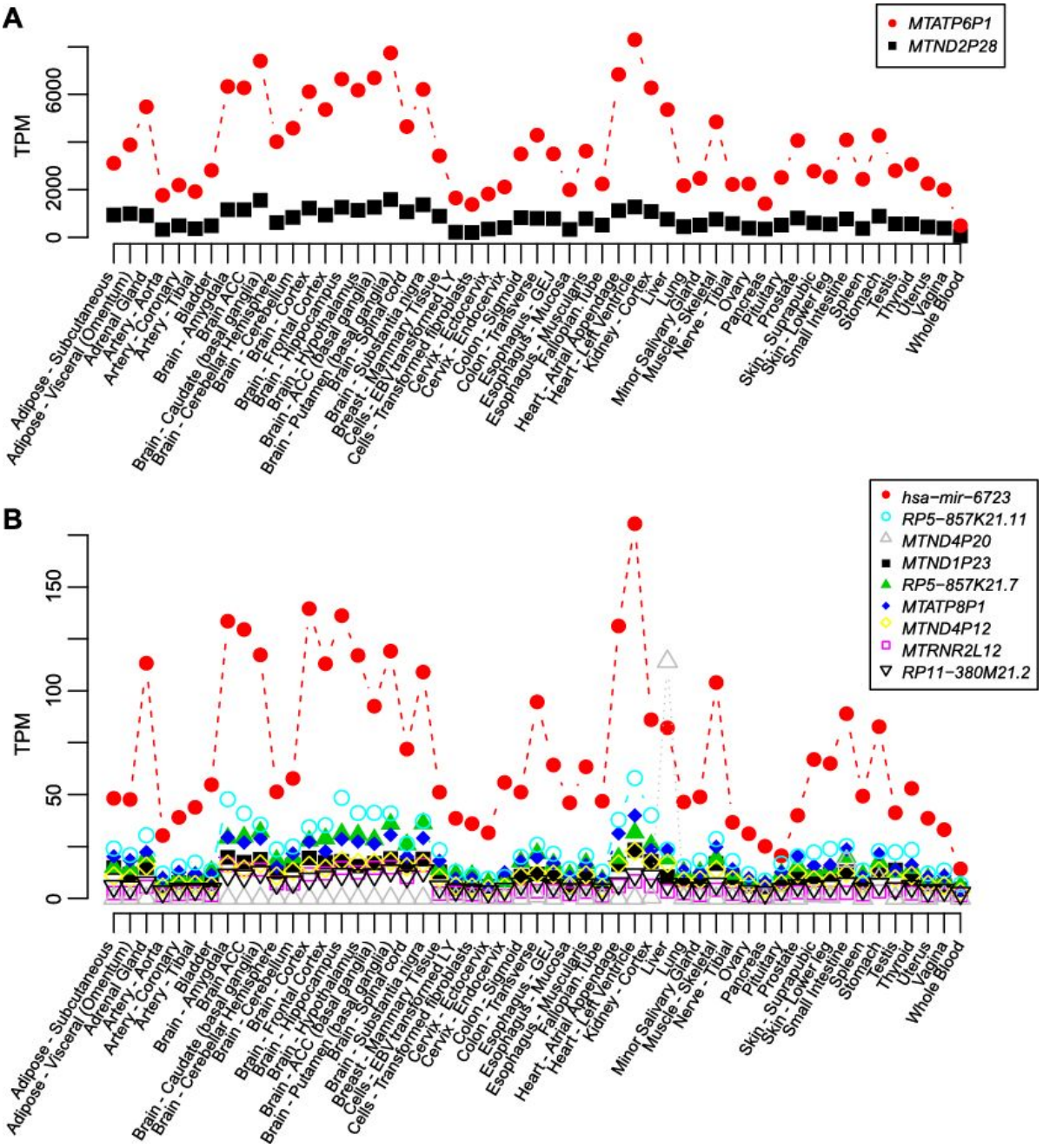

Figure 7: Medium and highly expressed genes. TPM quantification per tissue **(A)** >= 1000.0 TPM in at least, one tissue **(B)** between 10 and 1000 TPM.

In the Heatmap from Figure 6, we are including all genes that have at least one tissue with > 0.5 TPM. However, focussing on the more expressed genes (Figure 7), they have a clear tendency of expression depending on the tissue, being Heart and Brain the most expressed ones. These 11 genes expressed in > 10 TPM in at least one tissue are localized in 5 different NUMTs and except *MTRNR2L12,* classified as a protein-coding gene, the other 10 genes are identified as pseudogenes by ENSEMBL-Biotype (see Table 8). The table also includes the corresponding mitochondrial genes which sequence are included in the nuclear gene. As we can see, NUMT_01.001 contains the 7 genes more expressed.

Table 8. Genetic features of medium and high expressed genes. The table shows the corresponding NUMTs id, size, localization and the biotype classification for the nuclear-expressed genes with > 10 TPM in at least one tissue. Additionally, it is also indicated the corresponding mitochondrial gene which sequence are included in the NUMTs, its biotype classification and the % of the nuclear gene that includes each mitochondrial gene and the % of the mitochondrial gene that is present in the nuclear gene.

id	NUMT size	localization	Gene symbol (GTEx)	Nuclear gene biotype	Mito. gene	Mito. gene biotype	% overlap nuclear gene	% overlap mt. gene
NUMT_01.001	5843	intronic	*MTATP6P1*	pseudogene	*MT-ATP6*	protein_coding	99.85%	99.85%
NUMT_01.001	5843	intronic	*MTND2P28*	pseudogene	*MT-ND2*	protein_coding	99.81%	100.00%
NUMT_01.001	5843	intronic	*hsa-mir-6723*	pseudogene	*MT-CO1*	protein_coding	99.94%	100.00%
NUMT_01.001	5843	intronic	*RP5-857K21.11*	pseudogene	*MT-CO3*	protein_coding	99.82%	69.60%
NUMT_01.001	5843	intronic	*MTATP8P1*	pseudogene	*MT-ATP8*	protein_coding	99.51%	99.51%
NUMT_01.001	5843	intronic	*RP5-857K21.7*	pseudogene	*MT-CO2*	protein_coding	100.00%	99.71%
NUMT_01.001	5843	intronic	*MTND1P23*	pseudogene	*MT-ND1*	protein_coding	99.72%	100.00%
NUMT_05.030	5218	intronic	*MTND4P12*	pseudogene	*MT-ND6*	protein_coding	38.08%	100.00%
					MT-TE	Mt_tRNA	4.94%	100.00%
NUMT_18.003	202	intronic	*RP11-380M21.2*	pseudogene	*MT-CO2*	protein_coding	100.00%	75.51%
NUMT_10.029	862	intronic	*MTND4P20*	pseudogene	*MT-ND4*	protein_coding	96.05%	100.00%
NUMT_03.014	1322	partial_gene	*MTRNR2L12*	protein_coding	*MT-TV*	Mt_tRNA	7.02%	100.00%
					MT-RNR2	Mt_rRNA	71.59%	44.48%
					MT-RNR1	Mt_rRNA	21.18%	100.00%

Table 8 shows that *MTRNR2L12* is the only MTRNR2L member with >10 TPM detected in at least, one tissue. However, as we can see in Figure 8, it was detected the expression (> 0.5 TPM) of 8 genes classified as protein-coding genes, which includes the high expressed *MTRNR2L12* gene and *MTRNR2L1, MTRNR2L2, MTRNR2L4, MTRNR2L6, MTRNR2L8, MTRNR2L9* and *MTRNR2L10.* Comparing with the ones with GO term, only *MTRNR2L3, MTRNR2L5* and *MTRNR2L7* are missing in our expressed subset, since *MTRNR2L11* and *MTRNR2L13* are not in GO database.

For the other genes not classified as pseudogenes in Ensembl-biotype, *RP11-551L14.6* (lincRNA) is the only one detected as expressed (see Figure 8). *MIR4484* (miRNA), *RNU6-656P* (snRNA) have shown very low expressed levels (0 for most of the tissues) and the other annotated RNAs are not included in RNA-seq data from GTEx. It is possible that due to their small size, they were not detected by the RNA-seq analysis.

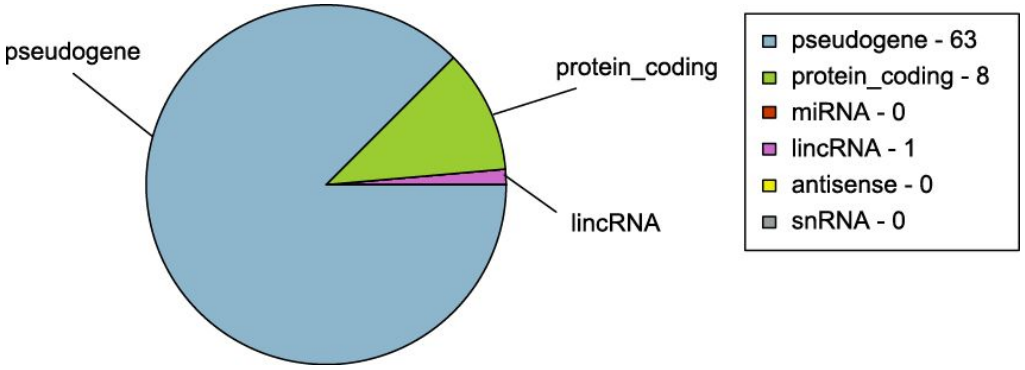

Figure 8. Ensembl-Biotype classification of expressed genes. Only 72 out of 452 genes with expression data in GTEx Portal are expressed (>= 0.5 TPM in at least one tissue). 62 out of these 72 genes are classified as pseudogenes in Ensembl-Biotype. For the other annotated genes, 8 corresponds to protein-coding genes, 1 lincRNA and any of the other annotated RNAs are expressed (miRNAs, antisense nor snRNA).

Focussing on the low, but expressed genes, some of the annotated genes includes the 100% of a mitochondrial tRNA (see Table 9). For example, the gene *MTND5P1*, which is originated mainly from the mitochondrial gene *MT-ND5* (which represents the 84.66% of the nuclear gene), also contains the full sequence of three additional mitochondrial genes (*MT-ND6, MT-TS2* and *MT-TL2*). Since in HGNC nomenclature of our total list of annotated Ensembl genes only includes two genes originated from mitochondrial tRNAs (*TRNAQ41P* in NUMT_17.007, which is not expressed, and *TRNAS30P* in NUMT_17.017, not included in GTEx dataset), maybe other tRNAs are not yet annotated as tRNAs or they are included as part of a bigger nuclear gene.

Table 9. Genetic features of four low expressed genes. As an example, the table shows the corresponding NUMTs id, size, localization and the biotype classification for the nuclear-expressed genes detected in GTEx with < 0.5 TPM in all tissues. Additionally, it is also indicated the corresponding mitochondrial gene which sequence are included in the NUMTs, its biotype classification and the % of the nuclear gene that includes each mitochondrial gene and the % of the mitochondrial gene that is present in the nuclear gene.

Gene symbol (GTEx)	Ensembl Biotype (nuclear gene)	Gene symbol (GTEx)	Ensembl Biotype (mt. gene)	% overlap nuclear gene	% overlap mito gene
MTND5P1	pseudogene	*MT-ND6*	protein_coding	38.13%	100%
		MT-ND5	protein_coding	84.66%	100%
		MT-TS2	Mt_tRNA	1.19%	100%
		MT-TL2	Mt_tRNA	11.96%	100%
RP11-750B16.1	pseudogene	*MT-CO1*	protein_coding	100%	99.52%
RP11-28B23.1	pseudogene	*MT-ATP6*	protein_coding	100%	28.29%
RP11-269G24.7	pseudogene	*MT-CO3*	protein_coding	100%	35.48%

3.4.2. Mitochondrial Genes

Additionally, since the BioMart Bioconductor package also includes mitochondrial chromosome, the mitochondrial genes were also included in our analysis to see its pattern of expression through different tissues included in GTEx RNA-seq data (see Figure 9).

At the expression level (online: File 16 “mean_tpm_GTExMITO.txt”), the 13 protein coding genes in mtDNA and the 2 rRNA are highly expressed, having all tissues an expression level of at least 500 TPM. However, in the case of tRNAs, some of them, are very low expressed or not expressed in some tissues. For example *MT-TQ, MT-TW* and *MT-TD*, which are 0 or < 0.5 in almost the 70% of the tissue tested. One of the most extreme cases is the ovary, where 13 tRNA are not expressed but the 15 mitochondrial proteins are highly expressed.

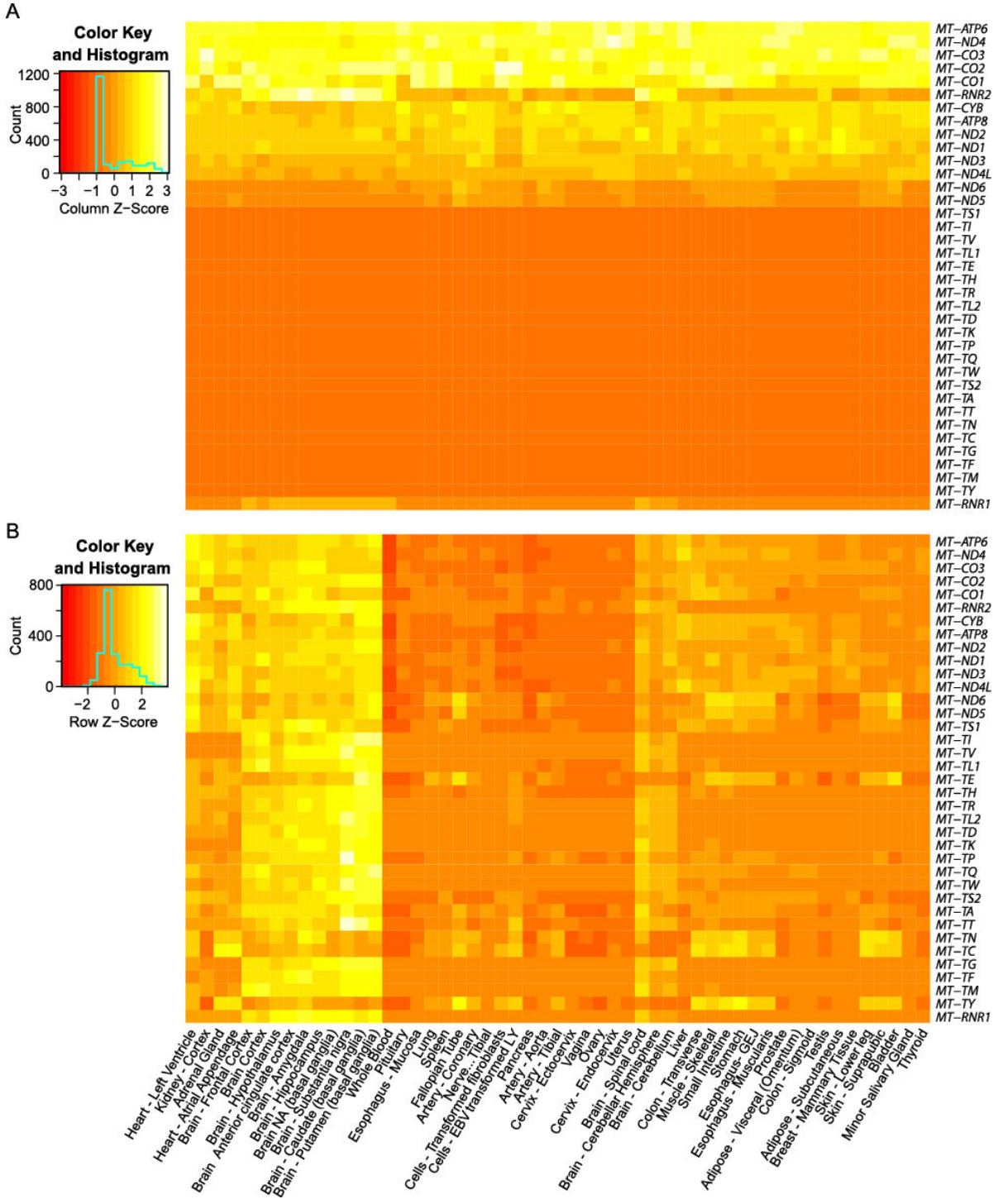

Figure 9. Heat map with the expression profile of all mitochondrial genes. **(A)** Z-score ($z = (x - \mu) / \sigma$) normalized by column, i.e. for each tissue independently, based on the expression profile of the different genes. **(B)** Z-score normalized by row, i.e. for each gene independently based on its expression profile in the different tissues.

4. Discussion

The results of this study show that NUMTs not only originate pseudogenes, they also give rise to expressed genes in different tissues. Because of this, we emphasise the importance of its correct annotation in Ensembl Biotype and the additional gene databases. However, if the nucleus machinery translates the nuclear-expressed genes, they probably encode proteins with a different structure than its original mitochondrial proteins. In fact, Ensembl Biotype classifies most of these genes as pseudogenes due to this reason. Nevertheless, some of these genes are expressed, and deeper wet experimental studies will be needed to describe the localisation of its mRNAs, and proteins to elucidate its new or conserve function. As it is mentioned before, the mtDNA is continuously transferred to the nucleus DNA, and as a consequence, the mtDNA has been reduced since its initial fusion with eukaryotic cell two billion years ago [15]. This reduction is in part, due to the posttranslational relocalization of proteins coding within NUMTs genes. However, it is reported in yeast cells that mRNA from the nucleus can be re-localised to different organelles, including the mitochondrial, to its posterior translation [19]. If this phenomenon is also present in human cells, some of the detected genes within NUMTs may conserve its original function. However, our results also show that in the mitochondria, most tRNA are not expressed but all its protein-coding genes and rRNA are highly expressed. This observation should not surprise us since some eukaryotic mitochondrial do not have any tRNA encoded in its mtDNA [20]. That is possible due to the mitochondrial tRNA importation from old NUMTs [21]. The similar expression profile of mitochondrial genes and high and medium nuclear-expressed genes may be is a reflexion of the number of mitochondria in the cells of highly expressed tissues (heart and brain, which have a high energy demand) together with the demand to import mRNA or proteins coding from the nucleus to the mitochondria. Additionally, we suggest that some of these nuclear-expressed genes could be post-transcriptionally processed and tRNA encoded in them, could be imported to the mitochondria.

Our results also highlight the importance of the genetic context where the mitochondrial origin DNA is inserted. For example, NUMT_01.001 seems to be inserted in a location where the expression is favoured since the 7 genes encoded in it, are the most highly expressed. Additional epigenetic studies will be needed to elucidate the characteristics of the chromatin in all insertions to show a possible correlation between expression and permissive or repressive histone marks or presence of DNA methylation.

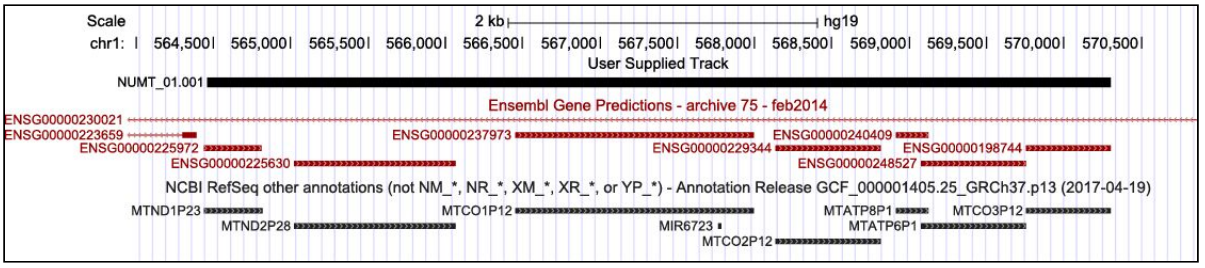

Figure 10. Visualising NUMT_01.001 in UCSC genome browser. Screenshot of the seven genes within NUMT_01.001 in UCSC genome browser (hg19).

The expression results we show are not complete. The fact that any annotated tRNA and miRNA were identified in GTEx as expressed may be due to the limitations of the RNA-Seq technique. Since document “Dynamic_Report_TFM.Rmd” contains all scripts needed to generate the expression tables automatically, once an updated version of RNA-seq means will be published, only by replacing the GTEx file, we can reanalyse our complete list of 456 genes.

5. Conclusiones

5.1. The general conclusion of the technical work

I. Bioconductor is a powerful tool to integrate different annotated gene data from different resources.

II. The continuous collaboration of all bioinformatic community not only in Bioconductor but also in all repositories give us the opportunity to perform high-throughput studies impossible to perform a few years ago.

III. The combination of all available data and in-home R-scripts give us the unique opportunity of downloading, filtering and classified in a relatively short-time big list of annotated features.

IV. The combination of R-studio and RMarkdown allows the automatic analysis of a completely new input (in our case, genomic coordinates) by the generation of a new set of documents (tables in .txt, figures or reports in .html or .pdf...) in a simple way for people unfamiliar with programming.

5.2. The general conclusion of the scientific results

I. Not all genes within NUMTs are pseudogenes since we describe expression levels of more than 0.5 TPM in at least, one human tissue for 72 of these genes.

II. The genetic context of the insertion influence the gene expression of genes within NUMTs since the first seven genes with higher level of expression are located within the same NUMT (NUMT_01.001).

III. It is needed the manual change of at least, 63 genes classified as “pseudogene” in the Ensembl Biotype since these genes are expressed.

IV. The differences in expression by tissue are similar between genes within NUMTs and mitochondrial genes.

V. NUMTs insertions contribute to nuclear transcription being potentially an active source of new gene variation and evolution of eukaryotic organisms.

5.3. Achieved objectives

The main objective of this study was to elucidate if NUMTs are originating pseudogenes and we demonstrate that not all of them does, some NUMTs are also originated from nuclear-expressed genes. Once we get the gene expression information we expand our analysis to additionally study mitochondrial gene expression and we compare both results. This analysis was not included in the first objectives, but since high and medium expressed genes coding within NUMTs shows a clear pattern of expression in the different tissues, we consider it important.

Additionally, all the specific objectives associated with bioinformatic analysis were achieved with some changes. The main technical objective was the creation of a complete list of annotated genes within our NUMTs together with different gene annotation, aim achieved during this work. The main change was not to use UCSC annotations but Ensembl annotations. We decided to focus on Ensembl annotated genes since are the ones associated with GTEx expression data.

Some objectives were also not achieved. Mainly, the Gene Ontology statistical analysis due to the low number of genes in our list included in Gene Ontology Consortium. We additionally wanted to perform a conservative and epigenetic analysis if we did not get any expressed gene but as we show, 72 genes in our list are express and the bioinformatic analysis then focused on expression.

5.4. Follow-up fo the working plan

In general, the working plan was correctly followed-up except for the expression analysis. I had some problems to access to RNA-seq GTEx data, and I spent more time than originally planned for it. Additionally, since the genes were not included in Gene Ontology Consortium, the time investment planned for that proposed were replace for a more in-depth analysis and classification of our final dataset.

5.5. Future work

After this first study and with the data recollected in our output table, exists a big list of future studies, by both, wet and dry experiments, to be performed.

First of all, by taking into consideration the representation of the origin mitochondrial gene, it could be proposed a description from those without gene symbol. For this propose and to better understand its functionality, other important in-silico analysis will need to study the protein structure of all expressed genes deeply. This information will give us a first idea of the potential function of this proteins. Additionally, wet experiment studying the localisation of this proteins will be needed.

Other future studies could be the better characterisation of the NUMTs insertions' genomic context. Especially, searching the epigenetic characteristics in the different tissues of NUMTs considering the gene position. A correlation between permissive histone marks and absence of DNA methylation in the corresponding gene promoters of the expressed genes but not in those tissues where the genes are silence give us the suggestion that the expression analysis from GTEx is complete. By observing these epigenetic characteristics in other parts of the NUMTs (and between 100 bp upstream and downstream the insertion) maybe suggest the existence of other genes not identified.

Finally, since these NUMTs are insertions from the actual human mitochondrial DNA, an evolutionary analysis to see its conservation in other primates and mammals will be needed to elucidate its temporal insertion.

6. Glossary

ATP: Adenosine triphosphate

bp: Base pair

DNA: Deoxyribonucleic acid

EMBL-EBI: European Molecular Biology Laboratory – European Bioinformatics Institute

GO: Gene Ontology

GTEx: Genotype-Tissue Expression

HGNC: HUGO Gene Nomenclature Committee

HUGO: Human Genome Organisation

lincRNA: Long non-coding RNAs

miRNA: Micro RNA

mRNA: Messenger RNA

mtDNA: Mitochondrial DNA

ncRNA: Non-coding RNAs

NUMT: Nuclear mitochondrial DNA sequences

RNA: Ribonucleic acid

RNA-seq: RNA sequencing

rRNA: Ribosomal RNA

snoRNA: Small nucleolar RNAs

snRNA: Small nuclear RNA

TPM: Transcripts Per Million)

tRNA: Transfer RNA

UCSC: University of California, Santa Cruz

7. Bibliography

[1] P. van den Boogaart, J. Samallo, and E. Agsteribbe, “Similar genes for a mitochondrial atpase subunit in the nuclear and mitochondrial genomes of neurospora crassa,” *Nature*, vol. 298, no. 5870, p. 187, 1982.

[2] T. Tsuzuki, H. Nomiyama, C. Setoyama, S. Maeda, and K. Shimada, “Presence of mitochondrial-dna-like sequences in the human nuclear dna,” *Gene*, vol. 25, no. 2, pp. 223–229, 1983.

[3] E. Richly and D. Leister, “NUMTs in sequenced eukaryotic genomes,” *Molecular biology and evolution*, vol. 21, no. 6, pp. 1081–1084, 2004.

[4] J. V. Lopez, N. Yuhki, R. Masuda, W. Modi, and S. J. O’Brien, “Numt, a recent transfer and tandem amplification of mitochondrial dna to the nuclear genome of the domestic cat,” *Journal of Molecular Evolution*, vol. 39, no. 2, pp. 174–190, 1994.

[5] G. Dayama, S. B. Emery, J. M. Kidd, and R. E. Mills, “The genomic landscape of polymorphic human nuclear mitochondrial insertions,” *Nucleic acids research*, vol. 42, no. 20, pp. 12640–12649, 2014.

[6] D. Bensasson, D.-X. Zhang, D. L. Hartl, and G. M. Hewitt, “Mitochondrial pseudogenes: Evolution’s misplaced witnesses,” *Trends in ecology & evolution*, vol. 16, no. 6, pp. 314–321, 2001.

[7] A. Goios, L. Prieto, A. Amorim, and L. Pereira, “Specificity of mtDNA-directed pcr—influence of nuclear mtdna insertion (numt) contamination in routine samples and techniques,” *International journal of legal medicine*, vol. 122, no. 4, pp. 341–345, 2008.

[8] D. Simone, F. M. Calabrese, M. Lang, G. Gasparre, and M. Attimonelli, “The reference human nuclear mitochondrial sequences compilation validated and implemented on the ucsc genome browser,” *BMC genomics*, vol. 12, no. 1, p. 517, 2011.

[9] D. Lascaro, S. Castellana, G. Gasparre, G. Romeo, C. Saccone, and M. Attimonelli, “The rhnumts compilation: Features and bioinformatics approaches to locate and quantify human numts,” *BMC genomics*, vol. 9, no. 1, p. 267, 2008.

[10] J. M. Kidd *et al.*, “Mapping and sequencing of structural variation from eight human genomes,” *Nature*, vol. 453, no. 7191, p. 56, 2008.

[11] A. Ramos *et al.*, “Nuclear insertions of mitochondrial origin: Database updating and usefulness in cancer studies,” *Mitochondrion*, vol. 11, no. 6, pp. 946–953, 2011.

[12] D. Mishmar, E. Ruiz-Pesini, M. Brandon, and D. C. Wallace, “Mitochondrial dna-like sequences in the nucleus (numts): Insights into our african origins and the mechanism of foreign dna integration,” *Human mutation*, vol. 23, no. 2, pp. 125–133, 2004.

[13] Huber, Wolfgang, et al. "Orchestrating high-throughput genomic analysis with Bioconductor." Nature methods 12.2 (2015): 115.

[14] Gentleman, Robert C., et al. "Bioconductor: open software development for computational biology and bioinformatics." Genome biology 5.10 (2004): R80.

[15] Lane, Nick, and William Martin. "The energetics of genome complexity." Nature 467.7318 (2010): 929.

[16] Stewart, James B., and Patrick F. Chinnery. "The dynamics of mitochondrial DNA heteroplasmy: implications for human health and disease." Nature Reviews Genetics 16.9 (2015): 530.

[17] Neupert, Walter, and Johannes M. Herrmann. "Translocation of proteins into mitochondria." Annu. Rev. Biochem. 76 (2007): 723-749.

[18] Schmidt, Oliver, Nikolaus Pfanner, and Chris Meisinger. "Mitochondrial protein import: from proteomics to functional mechanisms." Nature reviews Molecular cell biology 11.9 (2010): 655.

[19] Weis, Benjamin L., Enrico Schleiff, and William Zerges. "Protein targeting to subcellular organelles via MRNA localization." Biochimica et Biophysica Acta (BBA)-Molecular Cell Research 1833.2 (2013): 260-273.

[20] Adams, Keith L., and Jeffrey D. Palmer. "Evolution of mitochondrial gene content: gene loss and transfer to the nucleus." Molecular phylogenetics and evolution 29.3 (2003): 380-395.

[21] Schneider, André. "Mitochondrial tRNA import and its consequences for mitochondrial translation." Annual review of biochemistry 80 (2011): 1033-1053.

8. Supplementary information

In the next pages, the full content of “Dynamic-report-TFM.pdf” is shown, together with all scripts created to generate all datasets and graphs.

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:

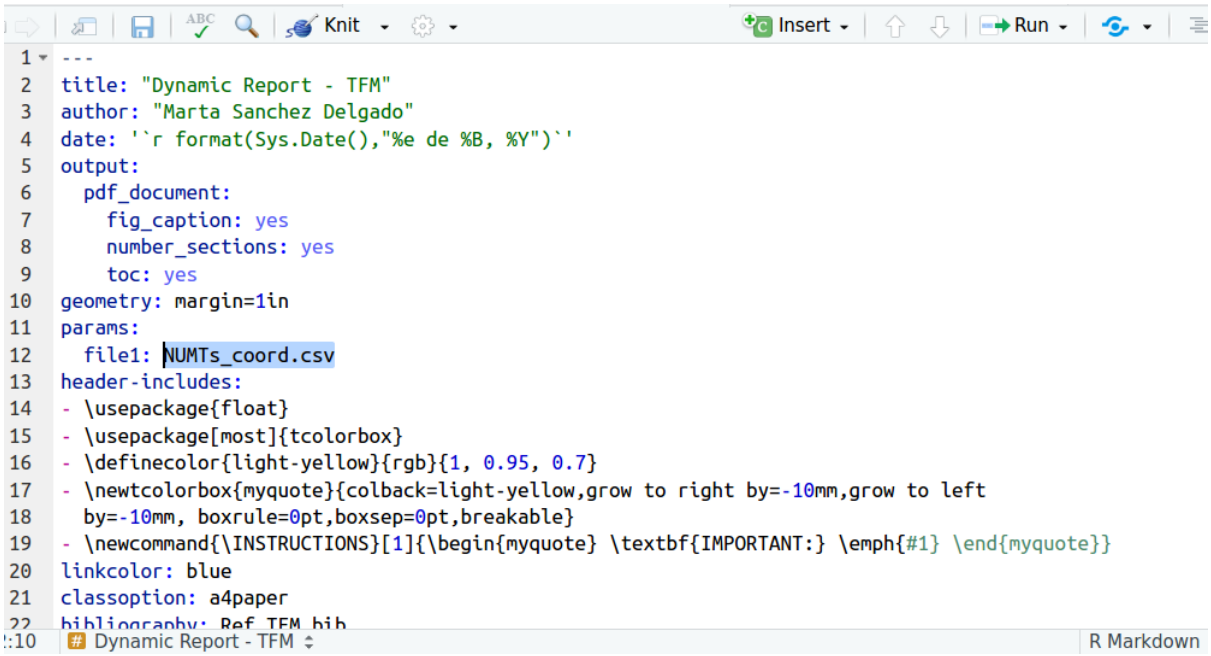

Figure 1: **Screenshot instruction to change the name of the 'input file'.** In the begining of the document you have the 'params' subsection, the 'file1:' corresponds to the 'input file' with the initial coordinates.

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.

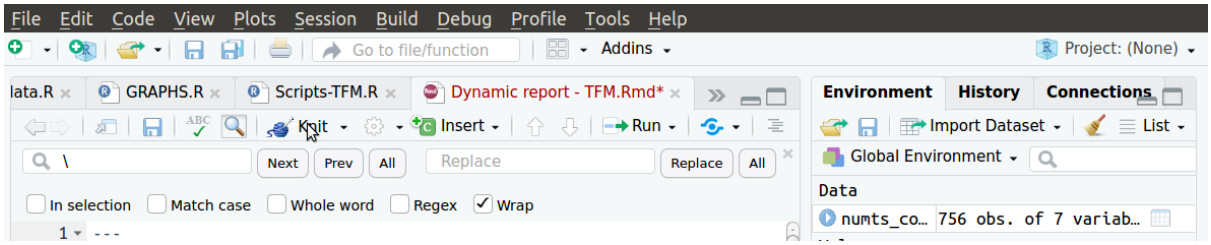

Figure 2: **Screenshot instruction for generate the new .html file.** In RStudio, by using R Markdown, we can generate the corresponding .html file by pression 'Knit' (in the upper-left part of the first square in RStudio).

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.

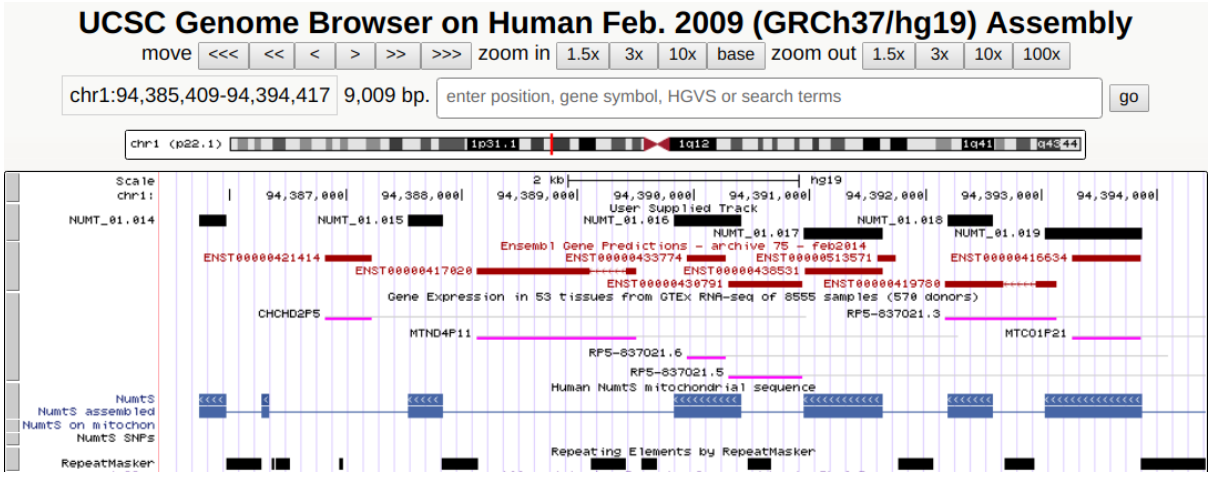

Figure 3: **Custom Track NUMNTs visualization in UCSC Genome Browser**. The first track on the picture is our Custom Track from the document "bed_NUMTsID.txt".

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.

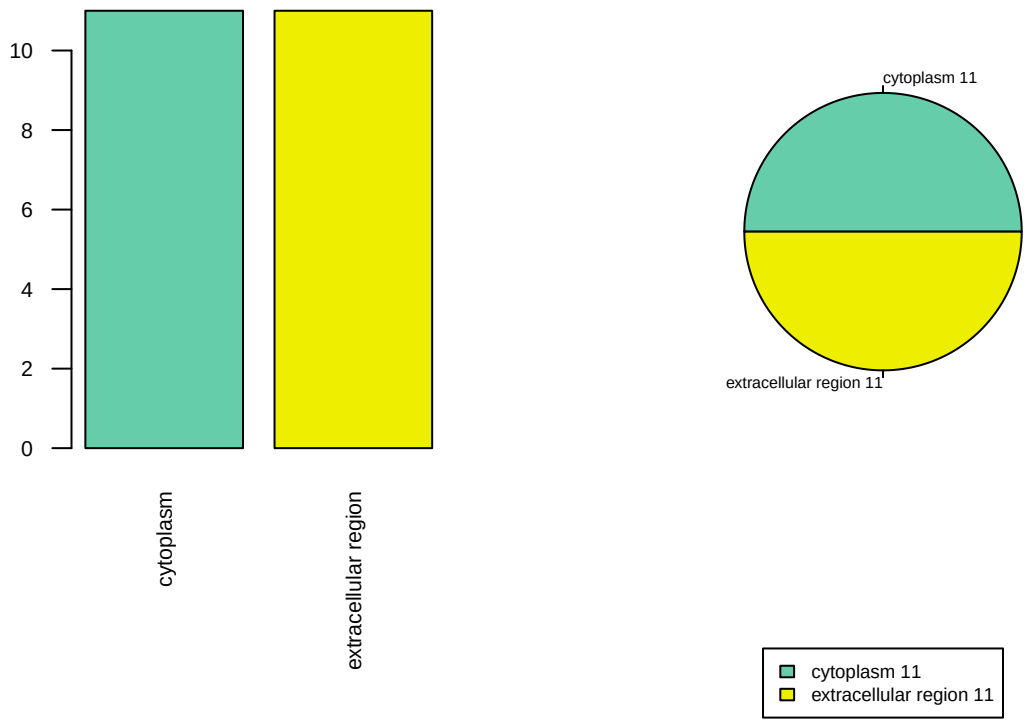

Figure 4: GO terms annotated for our list of genes.

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
```

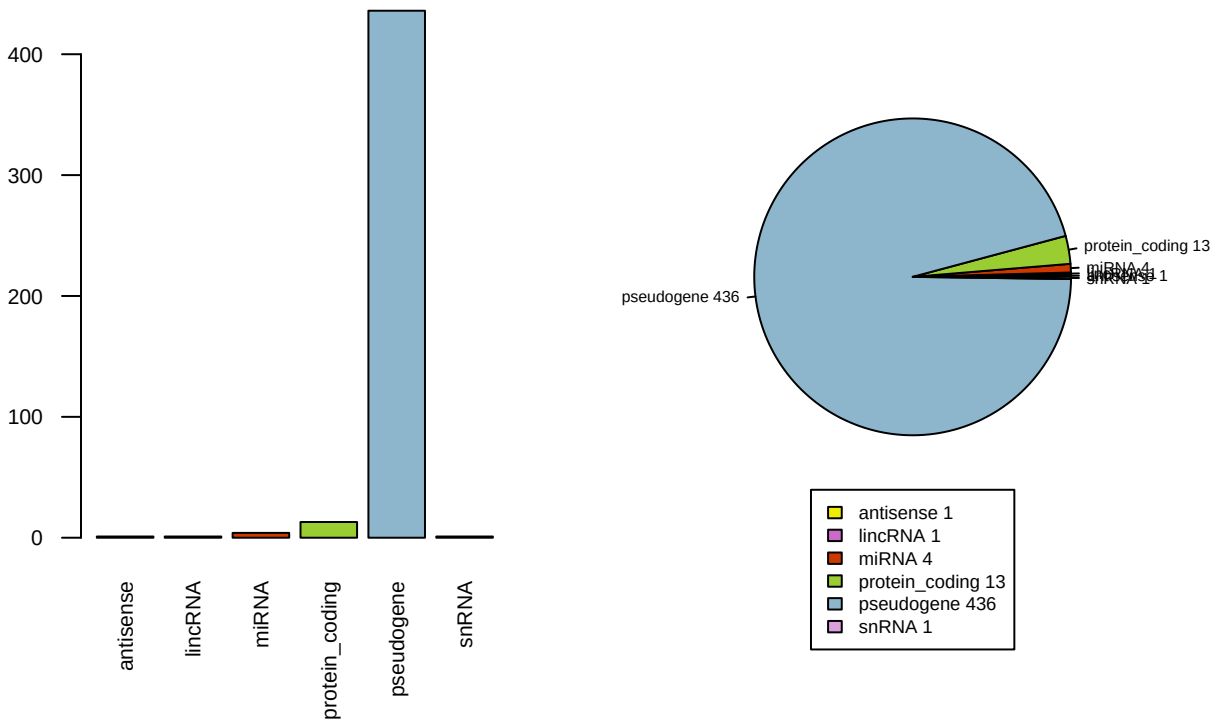

Figure 5: Gene biotype annotated for our list of genes.

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

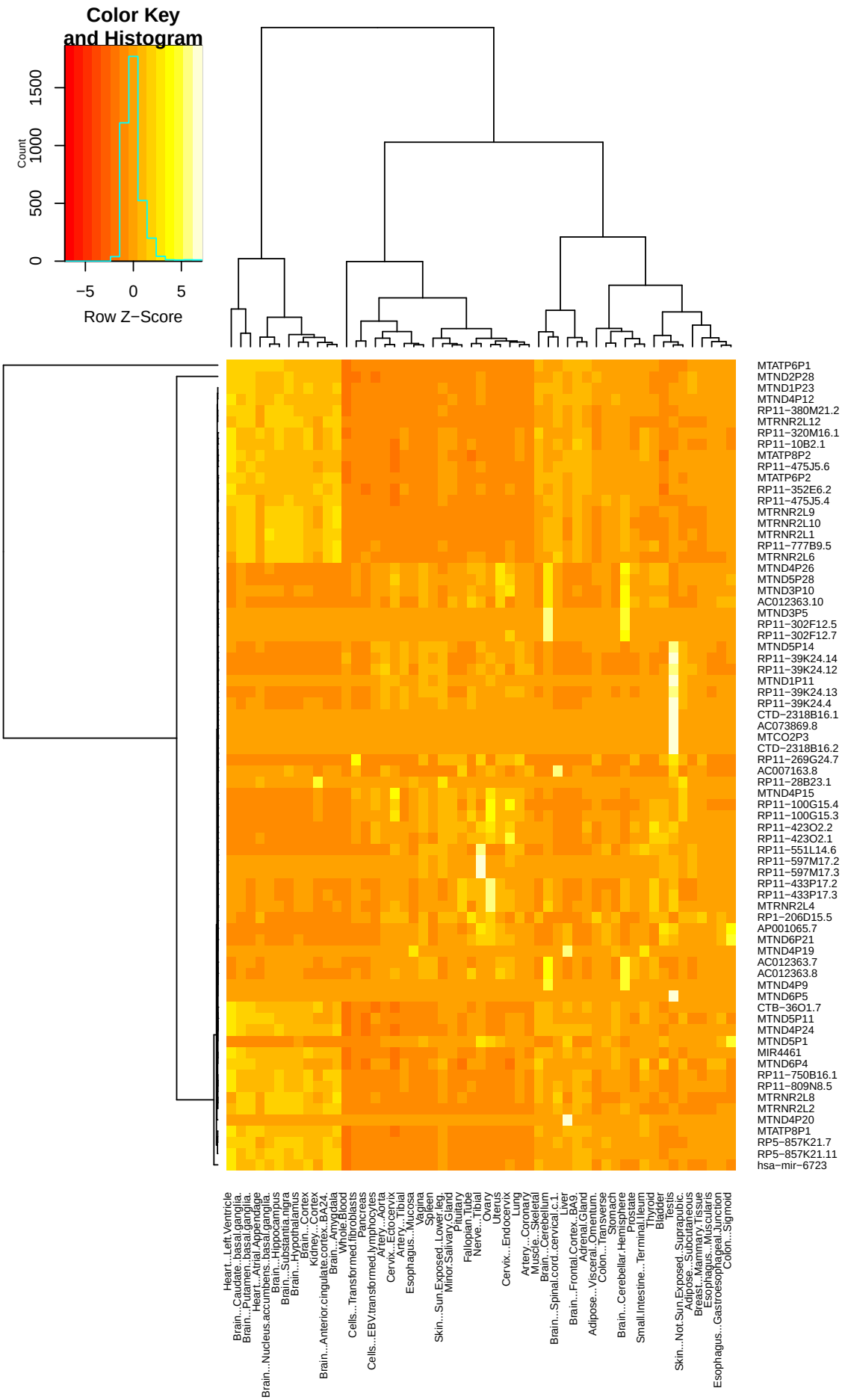

Figure 6: Heatmap of all expressed genes normalized by row (to see the different expression profile of each gene for the different tissues).

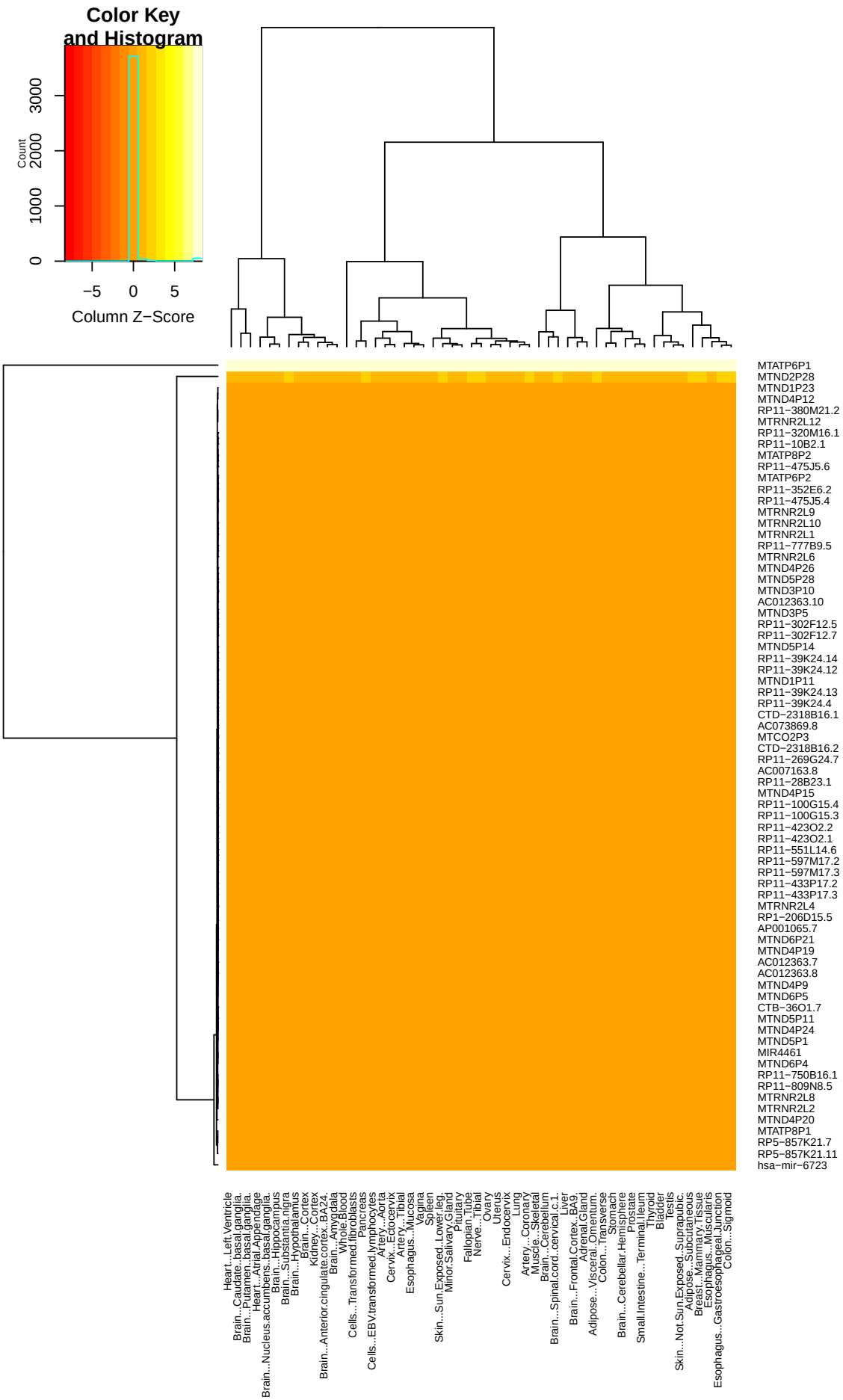

Figure 7: Heatmap of all expressed genes normalized by column (to see the different expression profile of the different gene for each tissue).

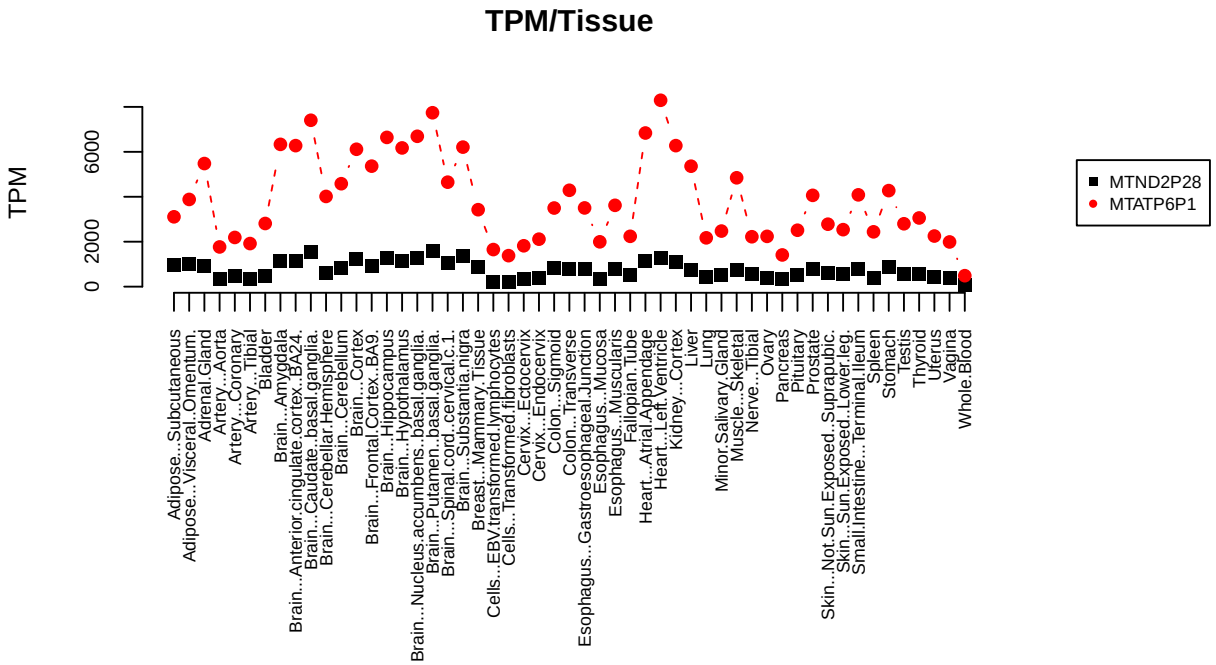

Figure 8: Graphical representation: expression profile of hight expressed genes (>=1000 TMP).

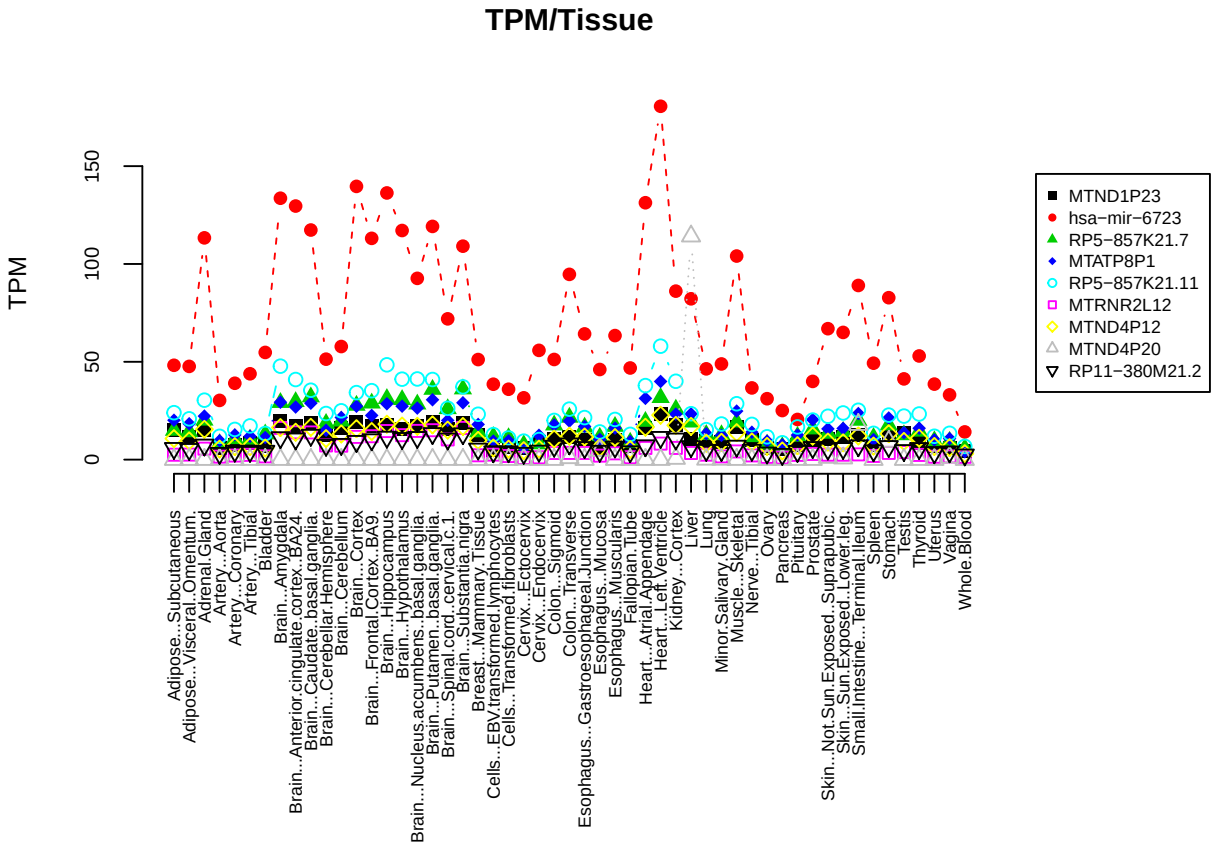

Figure 9: Graphical representation: expression profile of medium expressed genes (between 10 and 1000 TMP).

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
```

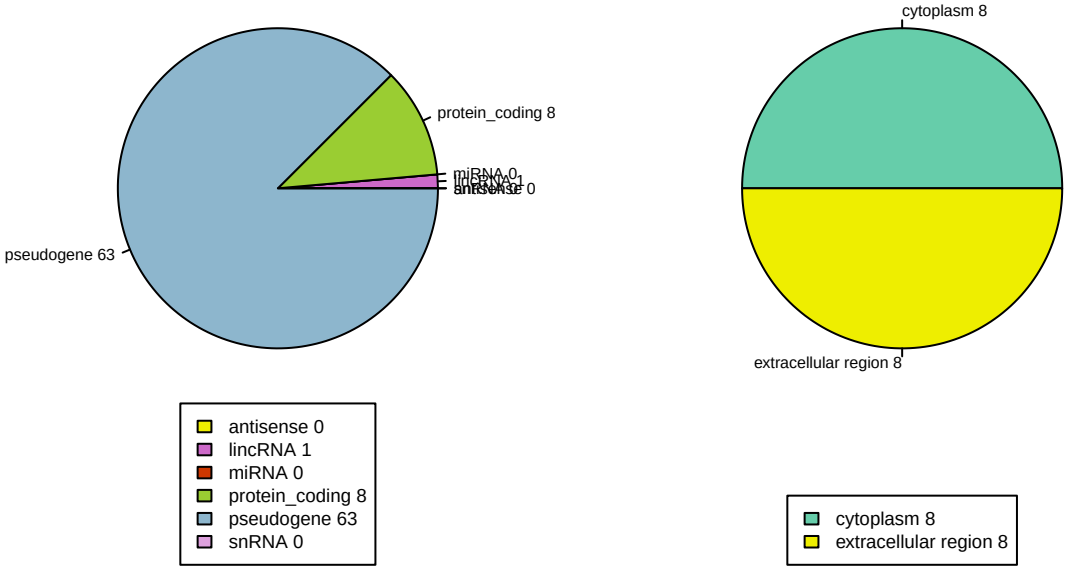

Figure 10: Gene biotype and GO term annotated for our list of expressed genes.

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)
```