

Exploration of a genomic annotations table (GTF)

DUBii 2019

Jacques van Helden, Hugo Varet and Frédéric Guyon

2019-01-23

Goal of the practical session

During this practical session, we will cover the following items:

1. Manipulate a table containing genomics data (*E. Coli* genome annotations).
2. Select a subset of the data/rows according to a given criterion.
3. Generate different graphical representations of these data.
4. Compute statistics describing the different types of annotations.

The GTF file format

The **GTF (General Transfer Format)** file format is extensively used to provide easily readable genomics annotations while being very handy with a computer.

Text files,

- ▶ one row per genomic “object” (gene, transcript, exon, intron, CDS, ...)
- ▶ one column per attribute (name, source, object type, genomic coordinates, description).

The GTF format is described on the following websites:

- ▶ <http://www.ensembl.org/info/website/upload/gff.html>
- ▶ <https://genome.ucsc.edu/FAQ/FAQformat.html#format4>

Find the GTF file of your favorite organism (on Ensembl)

1. Visit <http://ensemblgenomes.org/>.
2. Click on the link Bacteria.
3. Click on Download
4. In the **Filter** box, write *Escherichia coli*. The list of the proposed organisms is changing while you are writing.

For this session we will use the *E. Coli* GTF annotation file available here.

Page d'accueil d'EnsemblGenomes

<http://ensemblgenomes.org/>

The screenshot shows the Ensembl Genomes website homepage. The browser address bar displays 'ensemblgenomes.org'. The website header includes the Ensembl Genomes logo and navigation links: 'About us', 'Genomes', 'Data types', 'Data access', and 'FAQs'. A secondary navigation bar lists taxonomic groups: 'Bacteria', 'Protists', 'Fungi', 'Plants', 'Metazoa', and 'Vertebrates'. The main content area is divided into three columns. The left column features a purple banner with the text 'Ensembl Genomes: Extending Ensembl across the taxonomic space.' Below this, a section titled 'Ensembl Genomes Archive Sites Now Available' lists five archive sites with their respective URLs: 'http://ep40-metazoa.ensembl.org', 'http://ep40-protists.ensembl.org', 'http://ep40-fungi.ensembl.org', 'http://ep40-protists.ensembl.org', and 'http://ep40-bacteria.ensembl.org'. The right column contains two sections: 'What's New in Release 41 (September 2018)' and 'Have a question?'. The 'What's New' section includes links to 'Ensembl Bacteria' and 'Ensembl Fungi', with detailed text about updates to pathogen-host interactions and the addition of new fungal genomes. The 'Have a question?' section links to 'Frequently Asked Questions (FAQs)'. The bottom of the page includes a footer stating 'Ensembl Genomes is developed by EMBL-EBI and is powered by Ensembl software system for the analysis and visualisation of genomic data. For details of our funding'.

Ensembl Genomes: Extending Ensembl across the taxonomic space.

Ensembl Genomes Archive Sites Now Available

Ensembl Genomes now has a new set of archive sites from July 2018. These can be found at the following URLs.

<http://ep40-metazoa.ensembl.org>

<http://ep40-protists.ensembl.org>

<http://ep40-fungi.ensembl.org>

<http://ep40-protists.ensembl.org>

<http://ep40-bacteria.ensembl.org>

The archive sites allow researchers to access data from old releases via our web-based tools, and additionally will be able to display track hubs containing alignments and features located on older versions of genome assemblies that have since been upgraded in the live site. Archive sites will be searchable and BioMart will be available where they were produced for the site when live. Schema and API versions for archive sites will be the same as when the data was released.

posted 2017-11-07

Ensembl Genomes is developed by [EMBL-EBI](#) and is powered by [Ensembl](#) software system for the analysis and visualisation of genomic data. For details of our funding

What's New in Release 41 (September 2018)

[Ensembl Bacteria](#)

This release of [Ensembl Bacteria](#) contains updated annotations regarding pathogen-host interactions from [PHI-base](#).

[Ensembl Fungi](#)

This bumper release of [Ensembl Fungi](#) contains a fresh import of fungal genomes from the [ENA](#) and an updated gene set from [SGD](#) for [Saccharomyces cerevisiae](#), bringing the total number of genomes to 1023. Over 200 new fungal genomes have been integrated from ENA, ranging from the beautiful honey mushroom *Armillaria mellea* to *Fluccinria coronata*, a new and fast spreading pathogen of barley and oats. These genomes have been put through a revised pipeline that has annotated over 150 species with pathogen-host interaction data from [PHI-base](#). There is also an updated peptide comparison with a subset of 480 species chosen from across the fungal taxonomic groups and an updated pan-taxonomic comparison. The previous release of EnsemblFungi (40) has been archived at [ep40-fungi.ensembl.org](#).

[Ensembl Metazoa](#)

- Updated data
 - Updated BioMart
- New data

Have a question?

Frequently Asked Questions ([FAQs](#)) are now available for all domains of Ensembl Genomes. Have a question? Check if it's been asked before! If there is a FAQ missing, [contact us](#).

EnsemblGenomes Bacteria

<http://bacteria.ensembl.org/>

The screenshot shows the Ensembl Bacteria website in a web browser. The browser's address bar displays "bacteria.ensembl.org/index.html". The website's navigation bar includes links for "HMMER", "BLAST", "Tools", "Downloads", and "More". A search bar is located on the right side of the navigation bar.

The main content area is divided into several sections:

- Search for a gene**: A search box with the placeholder text "Search all species..." and a "Go" button. Below the box, an example search term "e.g. *ftsZ* or *uridine*" is provided.
- Search for a genome**: A search box with the placeholder text "Start typing the name of a genome...". Below the box, an example search term "e.g. type *esc* to find *Escherichia*" is provided.
- Search tips**: A list of instructions:
 - Search for a gene - type the name of a gene or other identifier into the search box above
 - Find a genome - click in the 'browse a genome' box above and start typing your genome name to find matching genomes
 - View full list of all Ensembl Bacteria species
 - Access Ensembl Bacteria programmatically
- What's New in Release 42**: A section titled "Genomes" with a list of updates:
 - 44,046 genomes (43,552 bacteria and 494 archaea) from 8244 species
- Did you know...?**: A section with a link to "REST service" and a note about accessing Ensembl Genomes data from any programming language.
- Come learn about Fungal Pathogen genomics!**: A section announcing a course on fungal pathogen genomics, scheduled for May 2019. It mentions that the course brings together bioinformaticians, biologists, clinicians, and computer scientists. The application and bursary deadline is 7th February 2019. A link to "details and registration" is provided.
- Archive site**: A section with a link to "Find all the data from release 40 of EnsemblBacteria at bacteria.ensembl.org/40 (July 2018)."
- Ensembl Bacteria**: A section describing the browser and its data sources. It states that the browser is a browser for bacterial and archaeal genomes, and that the data is taken from the databases of the International Nucleotide Sequence Database Collaboration (the European Nucleotide Archive at the EBI, GenBank at the NCBI, and the DNA Database of Japan).
- Non-redundant genomes**: A section describing the ENA's policy on non-redundant genomes. It states that the ENA houses over 90,000 prokaryotic genome assemblies, including multiple strains of many species. To reduce redundancy, the ENA has adopted a policy (as of release 35 - April 2017) of only loading in new sequences that are relatively non-redundant with the existing data set, according to the criteria of the UniProt Knowledgebase (UNIPROT).

EnsemblGenomes Fungi

<http://fungi.ensembl.org/>

The screenshot shows the Ensembl Fungi website in a web browser. The browser's address bar displays the URL `fungi.ensembl.org/index.html`. The website's header includes the Ensembl Fungi logo, navigation links for `HMMER`, `BLAST`, `BioMart`, `Tools`, `Downloads`, and `More`, along with a `Login/Register` link and a search bar. The main content area is divided into several sections:

- Search:** A search bar with the placeholder text "All species" and a "Go" button. Below it, an example search term "e.g. *NAT2* or *alcohol*" is provided.
- All genomes:** A list of links: "View full list of all Ensembl Fungi species" and "Edit your favourites".
- Favourite genomes:** A list of five genomes, each with a small icon and a label:
 - Saccharomyces cerevisiae* R64-1-1
 - Schizosaccharomyces pombe* ASM294v2
 - Aspergillus nidulans* ASM1142v1
 - Puccinia graminis* ASM14992v1
 - Magnaporthe oryzae* MG8
- Join the *Zymoseptoria tritici* community annotation project:** A text block describing the project and providing contact information for Marc-Henri Lebrun.
- Wellcome Genome Campus Advanced Course on Fungal Pathogen genomics:** A text block announcing the course and providing a link to details and registration.
- New archive site:** A text block providing a link to find data from the previous release of Ensembl Fungi.
- Ensembl Fungi:** A text block describing the browser and its data sources.

EnsemblGenomes Fungi Download page

<http://fungi.ensembl.org/info/website/ftp/>

FTP Download

Detailed information about the available data and file formats can be found [here](#).

The data can also be downloaded directly from the Ensembl Fungi [FTP server](#).

Database dumps

Entire databases can be downloaded from our FTP site in a variety of formats. Please be aware that some of these files can run to many gigabytes of data.

Looking for [MySQL dumps](#) to install databases locally? Instructions for loading MySQL dumps onto a local MySQL server can be found on the [Ensembl website](#).

Each directory on [ftp.ensemblgenomes.org](#) contains a [README](#) file, explaining the directory structure.

Programmatic data access

Data can be accessed programmatically in a number of ways, including the [REST service](#) and [Perl API](#). For full details see the [Programmatic access](#) documentation.

Multi-species data

Database	MySQL	TSV	EMF	MAF	XML
Pan_compara Multi-species	MySQL	TSV	EMF		XML
Fungal Multi-species	MySQL	TSV	EMF	MAF	XML
Ensembl Mart	MySQL				

Single species data

Show 10 5 entries

Species	DNA	cDNA	CDS	ncRNA	Protein	EMBL	GENBANK	MySQL	TSV	GTF	GFF3	GVF	VCF	VEP
<i>Saccharomyces cerevisiae</i>	FASTA (DNA)	FASTA (cDNA)	FASTA (CDS)	FASTA (ncRNA)	FASTA (protein)	EMBL	GenBank	MySQL (core) MySQL (other features) MySQL (transcript) MySQL (variation)	TSV	GTF	GFF3	GVF	VCF	VEP
<i>Saccharomyces cerevisiae_cen_pk113_7d_gca_000269885</i>	FASTA (DNA)	FASTA (cDNA)	FASTA (CDS)	FASTA (ncRNA)	FASTA (protein)	EMBL	GenBank	MySQL (core)	TSV	GTF	GFF3			VEP
<i>Saccharomyces cerevisiae_nc1118_gca_000218975</i>	FASTA (DNA)	FASTA (cDNA)	FASTA (CDS)	FASTA (ncRNA)	FASTA (protein)	EMBL	GenBank	MySQL (core)	TSV	GTF	GFF3			VEP
<i>Saccharomyces cerevisiae_fosterb_gca_000190255</i>	FASTA (DNA)	FASTA (cDNA)	FASTA (CDS)	FASTA (ncRNA)	FASTA (protein)	EMBL	GenBank	MySQL (core)	TSV	GTF	GFF3			VEP
<i>Saccharomyces cerevisiae_fosterro_gca_000326005</i>	FASTA (DNA)	FASTA (cDNA)	FASTA (CDS)	FASTA (ncRNA)	FASTA (protein)	EMBL	GenBank	MySQL (core)	TSV	GTF	GFF3			VEP
<i>Saccharomyces cerevisiae_gca_001634845</i>	FASTA (DNA)	FASTA (cDNA)	FASTA (CDS)	FASTA (ncRNA)	FASTA (protein)	EMBL	GenBank	MySQL (core)	TSV	GTF	GFF3			VEP

Define and create your working directory

Exercise: create a working directory named `workDir` in your home folder and go inside.

```
## Define the working directory
workDir <- "~/intro_R/explorer_un_GTF"

## Create the working directory
dir.create(workDir, recursive = TRUE, showWarnings = FALSE)

## Go to the working directory
setwd(workDir)
getwd()      ## Check your current location
list.files() ## List files (should be empty if just created)
```

Downloading the GTF file

Exercise: download the GTF file in the working directory (optionally, adapt the command to load a GTF of your interest). Before downloading the file we check if it is already present in the working directory. If yes, we skip the download.

****Tip:**** use the commands `file.exists`, `download.file`.

Downloading the GTF file: solution

```
## Define the URL of the file to download
gtf.url <- "ftp://ftp.ensemblgenomes.org/pub/bacteria/release
## create a directory to store the file
dir.create("data", showWarnings = FALSE)
## create a local filename
destfile <- paste0("data/", basename(gtf.url))
print(destfile)
```

```
[1] "data/Escherichia_coli_str_k_12_substr_mg1655.ASM584v2"
```

```
## Download the file, but only if not yet there
if (file.exists(destfile)) {
  message("GTF annotation file already there: ", destfile)
} else {
  message("Downloading GTF annotation file")
  download.file(url = gtf.url, destfile = destfile)
```

Loading a data table in R

Commands: `read.table`, `read.delim`, `read.csv`.

R includes several types of tabular structures (`matrix`, `data.frame`, `table`). The most widely used is `data.frame()`, which consists in a table of values with a type (strings, integer, ..) attached to each column, and names associated to rows and columns.

The function `read.table()` enables to read a text file containing tabular data, and to store its content in a variable.

Several finctions derived from `read.table()` facilitate the loading of different formats.

- ▶ `read.delim()` for files where a particular charcater is used as column separator (by default the tab character “`^`”).
- ▶ `read.csv()` for “comma-searated values”.

Loading the GTF file

Load the GTF file in a variable named featureTable.

****Tip:**** command `read.delim`.

```
## Load GTF file in a data.frame
featureTable <- read.delim(destfile, comment.char = "#", sep = ";",
                           header=FALSE, row.names = NULL)

## The GTF format has no header, but we can define it based on the column
names(featureTable) <- c("seqname", "source", "feature", "start", "end",
                         "score", "strand", "frame", "attributes")
```

Exploring the content of a data table

Immediately after having loaded a data table, check its dimensions.

```
dim(featureTable) ## Dimensions of the tbale
```

```
[1] 25979      9
```

```
nrow(featureTable) ## Number of rows
```

```
[1] 25979
```

```
ncol(featureTable) ## Number of columns
```

```
[1] 9
```

Checking heads and tails

Displaying the full annotation table would not be very convenient, since it contains tens of thousands of rows.

We can display the first rows of the file with the function `head()`, and the last rows with `tail()`.

```
## Display the 5 first rows of the feature table  
head(featureTable, n = 5)
```

```
## Display the 5 last rows of the feature table  
tail(featureTable, n = 5)
```


Viewing a table

If you are using the **RStudio** environment, you can display the table in a dynamic viewer pane with the function `View()`.

```
## In RStudio, display the table in a separate tab  
View(featureTable)
```

The `View()` function is interactive, so it should not be used in a script because it would perturbate its execution.

Selecting columns

The last column of GTF files is particularly heavy, it contains a lot of semi-structured information.

We can select the 8 first columns and display the 5 first rows of this sub-table.

```
## Column selection + head
head(featureTable[,1:8], n=5)
```

	seqname	source	feature	start	end	score	strand	frame
1	Chromosome	ena	gene	190	255	.	+	
2	Chromosome	ena	transcript	190	255	.	+	
3	Chromosome	ena	exon	190	255	.	+	
4	Chromosome	ena	CDS	190	252	.	+	
5	Chromosome	ena	start_codon	190	192	.	+	

```
## Equivalent: selecting subsets of rows and columns
```

Feature types

Exercise: the column *feature* of the GTF indicates the feature table.

- ▶ List the feature types found in the GTF
- ▶ Count the number of features per type, and sort them by decreasing values.

Tip: commands `unique`, `table` and `sort`.

```
## List the types of features
unique(featureTable$feature)
```

```
[1] gene          transcript exon          CDS          start_codon
Levels: CDS exon gene start_codon stop_codon transcript
```

```
## Count the number of features per type
sort(table(featureTable$feature), decreasing = TRUE)
```

Décompte par valeur

The `table()` function allows to count the frequency of each value in a qualitative variable:

```
## Count the number of features per chromosome
table(featureTable$seqname)
```

```
Chromosome
25979
```

```
## Count the number of features per strand
table(featureTable$strand)
```

```
    -    +
13246 12733
```

Contingency table

We can compute the number of combinations between two qualitative variables:

```
## Table with two vectors
table(featureTable$strand, featureTable$feature)
```

	CDS	exon	gene	start_codon	stop_codon	transcript
-	2129	2307	2277	2128	2128	2277
+	2012	2257	2220	2012	2012	2220

```
## Same result with a 2-column data frame
table(featureTable[, c("strand", "feature")])
```

	feature					
strand	CDS	exon	gene	start_codon	stop_codon	transcript
	2129	2307	2277	2128	2128	2277

Computing feature lengths

- ▶ Add a column with feature lengths.

Note about feature length computation (explain why) :

$$L = \text{end} - \text{start} + 1$$

```
## Add a column to the table with genes lengths  
featureTable$length <- featureTable$end - featureTable$start
```

Filtering rows based on a column content

The function `subset()` enables to select a subset of rows based on a filter applied to the content of one or several columns.

We can use it to select the subset of features corresponding to genes.

Selecting genes from the GTF table

- ▶ Select of genes from the GTF table and store them in a separate variable named `genes`.
- ▶ Compute summary statistics about gene lengths

Tip: commands `subset`, `summary`.

```
## Select subset of features having "CDS" as "feature" attribute
genes <- subset(featureTable, feature == "gene")
```

```
## Print a message with the number of genes
message("Number of genes: ", nrow(genes))
```

```
## Compute basic statistics on genes lengths
summary(genes$length)
```

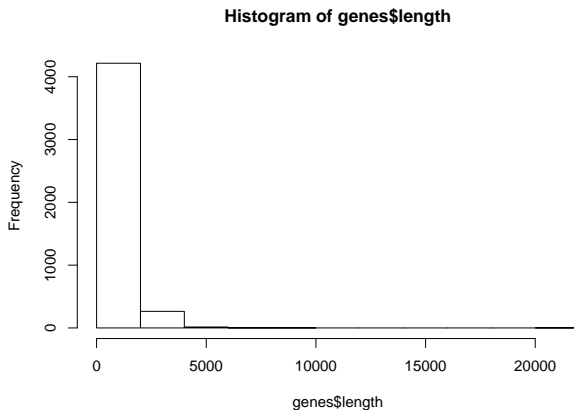
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
14 0	462 0	813 0	929 4	1221 0	21827 0

Exercices

1. Draw an histogram with gene length distribution. Choose a relevant number of breaks to display an informative histogram.
2. Draw a boxplot of gene lengths per strand. Are gene longer on the minus or plus strand?

Gene length histogram

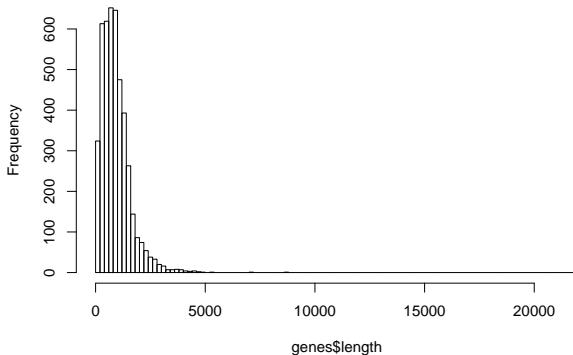
```
hist(genes$length)
```



Setting a relevant number of breaks

```
## Take more or less 100 bins  
hist(genes$length, breaks = 100)
```

Histogram of genes\$length



Gene length distribution – improving the output

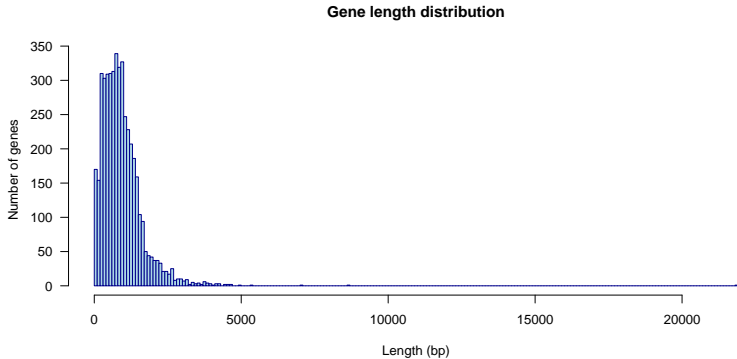


Figure 1: Distribution of gene lengths for E. Coli.

Gene length box plot

Other types of plots allow to explore the distribution of some data. In particular, boxplots display the median, the first and third quartiles and outlier values.

```
boxplot(length ~ strand, data = genes, col="palegreen", horizontal=
        las=1, xlab="Gene length", ylab="Strand")
```

