Data management and processing instructions

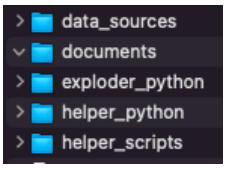
Important: set a value for *$rootRG* as described immediately below, in part a, before following any examples.

Commands shown in this document have been tested using MacOs `zsh` Examples use the double-quote symbol for decoding `$` values eg: *"$locus"* and *"$rootRG"*. Beware unintentional substitutions of this symbol for the double-quote symbols “ or ” which cause errors in `sh` interpretation.

Adding a new locus to the data management hierarchy

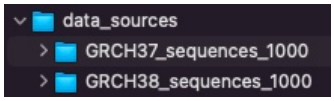
a) Locate and define a `$rootRG` directory.

The Github repository https://github.com/snowlizardz/rg_exploder_shared, has four directories at `$rootRG` level




```
caryodonnell@MacBook-Air Replicon % pwd
/Users/caryodonnell/Desktop/Replicon
caryodonnell@MacBook-Air Replicon % ls
data_sources   exploder_python       helper_python  helper_scripts make_rootRG.sh
caryodonnell@MacBook-Air Replicon % export rootRG="`pwd`"
caryodonnell@MacBook-Air Replicon % echo $rootRG
/Users/caryodonnell/Desktop/Replicon
caryodonnell@MacBook-Air Replicon %
```

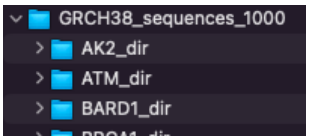
b) In `"$rootRG"/data_sources` these are the first-level data-source directories




```
caryodonnell@MacBook-Air Replicon % ls "$rootRG"/data_sources
GRCH37_sequences_1000  GRCH38_sequences_1000
...
```

`GRCH37_sequences_1000`**:** holds data files for build GRCH37
`GRCH38_sequences_1000`: holds data files for build GRCH38

c) Within each of these, there is one data directory for each locus:



Eg: **GRCH38_sequences_1000/AK2_dir** holds downloaded AK2 data from Ensembl; initially as a file called **ensembl.txt.gz,** and later processed versions of these files (see ‘sequence file processing’ below)

c) To add a new `"$locus"` called AK22, create a new folder in the appropriate directory

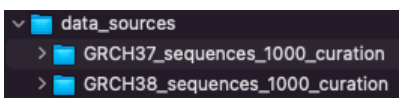
`GRCH38_sequences_1000/"$locus"_dir` **or** `GRCH37_sequences_1000/`*`$"locus"`*`_dir`

```
> locus="AK22"
> mkdir "$rootRG"/data_sources/GRCH38_sequences_1000/"$locus"_dir
```

d) Follow “Downloading a sequence file ...” instructions below for the new `"$locus"`

e) Follow “Automated data processing…” instructions below for the new `"$locus"`

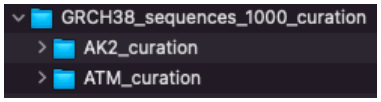
f) Also present in `"$rootRG"/data_sources` are the two main data-curation directories




```
caryodonnell@MacBook-Air Replicon % ls "$rootRG"/data_sources
GRCH37_sequences_1000          GRCH38_sequences_1000
GRCH37_sequences_1000_curation       GRCH38_sequences_1000_curation
...
```

`GRCH37_sequences_1000_curation`: these hold curated files for build GRCH37
`GRCH38_sequences_1000_curation`: these hold curated files for build GRCH38

g) Within each of these, there is one data directory for each locus, eg:



h) As with step c, add a new `$locus` folder in the appropriate curation directory eg: `GRCH38_sequences_1000_curation/"$locus"_curation`

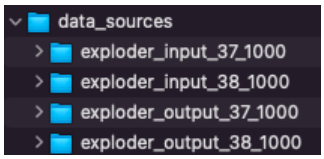
```
> locus="AK22"
> mkdir "$rootRG"/data_sources/GRCH38_sequences_1000_curation/"$locus"_dir
```

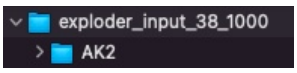
i) Follow “Maintaining the curation data” instructions below for the new `"$locus"`

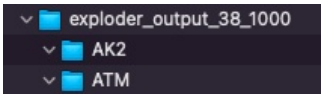
j) Follow “Mashup time” instructions below to include the new `"$locus"` data in an updated version of the lookup file `loci.json` in

```
"$rootRG"/data_sources/exploder_input_38_1000/loci.json
```

k) Within `"$rootRG"/data_sources` are further directories to hold the input & output data when running the application. Create a directory for each locus in both the input and output directories:







l) In `"$rootRG"/data_sources/exploder_input_38_1000/"$locus"` create soft links to files in the `GRCH38_sequences_1000_curation/"$locus"_curation` directory.

m) Locate `"$rootRG"/data_sources/exploder_python`
Create soft links to the desired input and output directories

```
caryodonnell@MacBook-Air exploder_python % pwd
/Users/caryodonnell/Desktop/Replicon/exploder_python
caryodonnell@MacBook-Air exploder_python % ls -l
…

input -> /Users/caryodonnell/Desktop/Replicon/data_sources/exploder_input_38_1000/
output -> /Users/caryodonnell/Desktop/Replicon/data_sources/exploder_output_38_1000/
```

Use `"$rootRG"/data_sources/helper_scripts/switch_links.sh` to flip quickly between different sets; 37 and 38, for example

o) To create the lookup file `"$rootRG"/data_sources/exploder_python/input/config.json`
Run the python script `RG_exploder_globals_make.py` (check values in `set_config_consts`) in the `exploder_python` directory:

```
> cd "$rootRG"/exploder_python/
> python3 RG_exploder_globals_make.py
```

p) Finally, run the application
`"$rootRG"/data_sources/exploder_python/RG_exploder_gui.py`

```
> cd "$rootRG"/exploder_python/
> python3 RG_exploder_gui.py
```

If the curation directory seems like overkill, it allows the maintenance of individual curated haplotype definitions, thereby separating the download-processed files from the input files. This is best illustrated by the EGFR set:

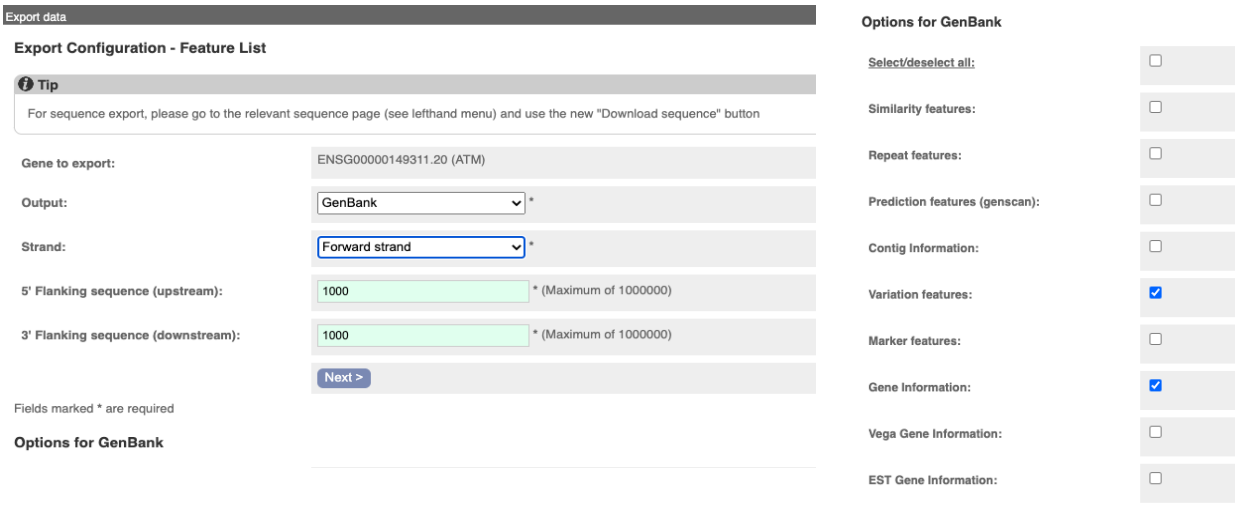
```
caryodonnell@MacBook-Air EGFR % pwd
/Users/caryodonnell/Desktop/Replicon/data_sources/
exploder_input_38_1000/EGFR
caryodonnell@MacBook-Air EGFR % ls -l
total 0
EGFR_000000.gb -> EGFR_curation/EGFR_noseq.gb
EGFR_curation ->
../../GRCH38_sequences_1000_curation/EGFR_curation/
EGFR_locseq.gb -> EGFR_curation/EGFR_locseq.gb
EGFR_p1.gb -> EGFR_curation/EGFR_p1.gb
EGFR_p1som1.gb -> EGFR_curation/EGFR_p1som1.gb
EGFR_p1som15.gb -> EGFR_curation/EGFR_p1som15.gb
EGFR_p1som32.gb -> EGFR_curation/EGFR_p1som32.gb
EGFR_p2.gb -> EGFR_curation/EGFR_p2.gb
EGFR_p2som14.gb -> EGFR_curation/EGFR_p2som14.gb
EGFR_p2som16.gb -> EGFR_curation/EGFR_p2som16.gb
EGFR_p2som39.gb -> EGFR_curation/EGFR_p2som39.gb
EGFR_transcripts.json ->
EGFR_curation/EGFR_transcripts.json

caryodonnell@MacBook-Air EGFR_curation % pwd
/Users/caryodonnell/Desktop/Replicon/data_sources/
exploder_input_38_1000/EGFR/EGFR_curation
caryodonnell@MacBook-Air EGFR_curation % ls -l
total 80
EGFR_dir -> ../../GRCH38_sequences_1000/EGFR_dir/
EGFR_ensembl -> EGFR_dir/EGFR_ensembl
EGFR_locseq.gb -> EGFR_dir/EGFR_locseq.gb
EGFR_noseq.gb -> EGFR_dir/EGFR_noseq.gb
EGFR_p1.gb
EGFR_p1som1.gb
EGFR_p1som15.gb
EGFR_p1som32.gb
EGFR_p2.gb
EGFR_p2som14.gb
EGFR_p2som16.gb
EGFR_p2som39.gb
EGFR_transcripts.json
```

Downloading a sequence file for a locus from Ensembl

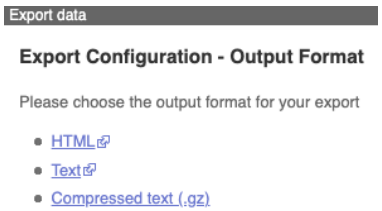
These instructions are suitable for downloading a new sequence, or when updating an existing one.

Starting at https://www.ensembl.org/Homo_sapiens/Info/Index:

- Find the gene of interest using Search & go to the Summary eg: ATM
  - Use the chosen gene name as *locus* below.
- Click on "export data" (LH menu)
- Select output: Flatfile/Genbank
- Select Forward Strand (preferred)
  - Alternatives are:
    - Feature Strand (This will be Forward or Reverse depending on the transcript)
    - Reverse Strand
- In "5' Flanking sequence (upstream)" and "3' Flanking sequence (downstream)": enter **1000**
  - A minimum value of 1000 is essential for supporting 'paired end reads'
- In "Options for Genbank":
  - Deselect all
  - Reselect: "variation features" and "gene information" (exon, mRNA & CDS definitions)
- Press "Next"



- In the new "Export data" window, click the "compressed text (gz)" link
- The downloaded file is named **ensembl.txt.gz**
  - Move this file, into a data directory called `"$rootRG"/data_sources/"$locus"_dir` eg: the `GRCH38_sequences_1000/AK2_dir` example above
  - Rename it to `"$locus"_ensembl` eg: `AK2_ensembl`

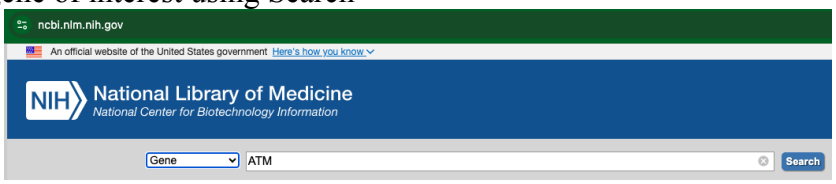


Downloading a sequence file for a locus from NCBI

These instructions are suitable for downloading a new sequence, or when updating an existing one

Starting at https://www.ncbi.nlm.nih.gov/

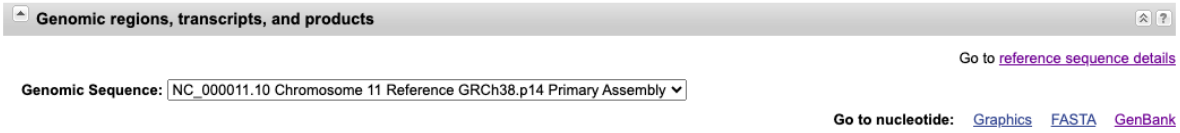
- Find the gene of interest using Search



- Click on the link in the gene card

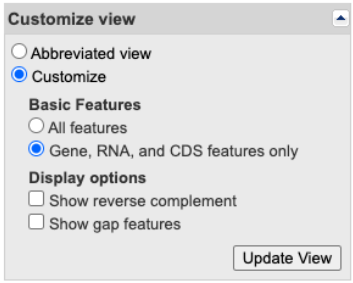
- Which shows the gene summary

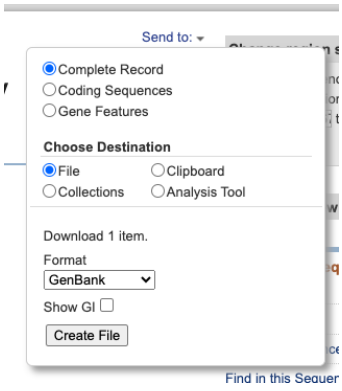
- Click on "Genomic regions, transcripts, and products", then scroll down



- Click on "Go to nucleotide ... Genbank" eg:
  - `https://www.ncbi.nlm.nih.gov/nuccore/NC_000011.10?report=genbank&from=108223067&to=108369102`
- Subtract 1000 from the start and add 1000 to the end:
  - `https://www.ncbi.nlm.nih.gov/nuccore/NC_000011.10?report=genbank&from=108222067&to=108370102`

- Top of page; modify "Customize view"; pulldown "Send to"; click "File" & "Create File"





- The output is a file called sequence.db

Automated data processing for downloaded Ensembl data

Introduction to data processing

There are two main objectives:

a) Create a `config.json` file as a lookup list for the application.

b) Remove, from the downloaded **ensembl.txt.gz**, all the data unnecessary for this application.

Adding new data into the `config.json` file could be done manually by looking at the existing examples. The helper scripts automate the filtering, `config.json` generation, and other fiddly bits.

Using a feature-filter Python script

A Python script, `"$rootRG"/helper_python/`*`embl_feature_filter_revise.py`* should be used to process a `"$locus"_ensembl` file in either gz or uncompressed format eg:

```
> cd "$rootRG"/data_sources/GRCH38_sequences_1000/"$locus"_dir
> python3 "$rootRG"/helper_python/embl_feature_filter_revise.py -i "$locus"_ensembl -a
```

The output files used by the `RG_exploder` application, are:

`"$locus"``_locseq.gb`: Contains a cleaned-up feature table: retaining only minimal db_xref identifiers; mRNA and CDS join data. Also holds the DNA sequence.

`"$locus"``_noseq.gb`: Contains **no** DNA sequence and the bare minimum definition data. In the application it is used to define the *`"$locus"`*`_000000` haplotype. This file is also used as a template for defining the variants in haplotypes, in the curation directory.

`"$locus"``_transcripts.json`: Holds lookup data for the GUI part of the application.

Other output files, useful for curation and checking:

`"$locus"``_filtered.gb`: The 'original file' with all the unwanted data taken out; same content as *`"$locus"`*`_locseq.gb`, but including all the variation features from the original source.

`"$locus"``_filtervar.gb`: As for *`"$locus"`*`_noseq.gb`, but including all the variation features. These may be useful for extracting a subset of variation features to make haplotype definition files.

Parameters for `embl_feature_filter_revise.py`:

`-i` is the downloaded-from-Ensembl input file eg: ensembl.txt.gz,

`-a` is necessary to produce "all" output files (described below)

`-j` *omits* mRNA and CDS *join* data in `"$locus"_transcripts.json`

- `-j` may be used **only** for the Python GUI; the browser version ***requires*** the join data

An alternative feature-filter Python script

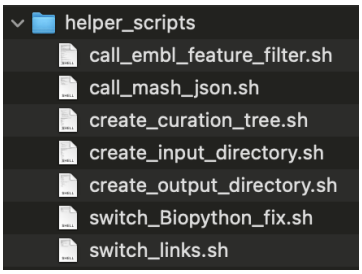
The Python script, `"$rootRG"/helper_python/`*`embl_feature_filter8.py`* does not use Biopython, using a line-by-line text elimination of 'dbxref= *database*' lines

eg:`unwanted=['xref="RefSeq_mRNA_predicted:', ... 'db_xref="RefSeq_ncRNA_predicted:']`

Note that new, unnecessary data appears over time, so this list needs to be maintained. This is a harder script to maintain than *`embl_feature_filter_revise.py`* but was developed first. It may be useful to compare the output between these two.

Filtering multiple data sets at once

To automate this one step further, by (re-)processing multiple locus directories at once, using scripts in `"$rootRG"/helper_scripts/`



Automating feature-filter scripts

Use the script `call_embl_feature_filter.sh` to run a feature_filter script repeatedly for each subdirectory of downloaded data.

a) Go to the directory where

```
> cd "$rootRG"/data_sources/GRCH38_sequences_1000/
```

Or

```
> cd "$rootRG"/data_sources/GRCH37_sequences_1000/
```

```
caryodonnell@MacBook-Air AK2_dir % cd "$rootRG"/data_sources/GRCH38_sequences_1000/
caryodonnell@MacBook-Air GRCH38_sequences_1000 % ls -l
total 0
AK2_dir
ATM_dir
…
STK11_dir
TP53_dir
caryodonnell@MacBook-Air GRCH38_sequences_1000 %
```

Then:

```
> sh "$rootRG"/helper_scripts/call_embl_feature_filter.sh
```

As supplied the default is to use *`embl_feature_filter_revise.py`*

Within `call_embl_feature_filter.sh` you may change which of the feature-filter scripts is used

```
# There are two filter programs ...
python_filter1="$rootapplicationdir$filter8
# embl_feature_filter8.py is a difficult-to-follow line-by-line parsing of the input file
python_filter2="$rootapplicationdir$filter_revise

...
# ... Pick one:
# python_filter=$python_filter1
python_filter=$python_filter2
```

Amendments for attention in these python scripts prior to execution

When adding a new locus, corresponding identifiers need to be added to the two dictionary lists within *embl_feature_filter_revise.py* and *embl_feature_filter8.py*

- `MANE_Select_dict` and `LRG_id_dict`
  - Each item in `MANE_Select_dict` is the transcript identifier assigned MANE_Select status for each locus.

```
MANE_Select_dict={
"AK2":"672715",
"ATM":"675843",
…
"STK11":"326873",
"TP53":"269305"
}
```

  - Likewise items in `LRG_id_dict` are the LRG identifiers for each locus.

```
LRG_id_dict={
"AK2":"133",
"ATM":"135",
…
"STK11":"319",
"TP53":"321"
}
```

  - A link to LRG was an early user-requirement for a link from “Locus”, but LRG now appears to be being phased out. Links now go to the LRG page in Ensembl.
  - To include the correct date requires a code change in `make_transcriptconstants()`, eg: `Release="Ensembl Release 105 (Dec 2021)"`

Data processing for downloaded NCBI data

Automated processing is currently **not** available for data downloaded directly from NCBI

There are significant differences between the content of Ensembl & Genbank downloads. Listing these, incompletely, here:

- There is no link between Transcript ID and CDS in Genbank output, unlike Ensembl
- The CDS in Genbank are clustered together in Genbank, and not interlaced with mRNA, as in Ensembl.
- There are far fewer tags that would need excluding
  - The major things to remove are:
    - Any information from genes not in the intended gene set (ie: overlaps or other-strand)
    - /translation="
- The terms for the transcript id and gene_id differ between Ensembl & Genbank eg:

Ensembl

```
gene            1001..147059
                  /gene=ENSG00000149311.20
                  /locus_tag="ATM"
                  /note="ATM serine/threonine kinase [Source:HGNC
                  Symbol;Acc:HGNC:795]"
mRNA ...
           /gene="ENSG00000149311.20"
     /     standard_name="ENST00000675843.1"
```

Genbank

```
gene          1001..147036
                  /gene="ATM"
                  /db_xref="GeneID:472"
                  /db_xref="HGNC:HGNC:795"
 mRNA ...
           /gene="ATM"
           /transcript_id="XM_011542840.4"
```

Advantage:
a) Far less removal of unwanted content is required

Disadvantages:
a) Transcript ID tags are different
b) Cannot link a CDS to a given transcript with the information in this file, it would have to be inferred by order; a complication being the potential inclusion of non-coding mRNA

Automation of post-processing

Introduction to post-processing

Three more directories are required for data maintenance:

Curation: `"$rootRG"/data_sources/GRCH38_sequences_1000_curation`
Input: `"$rootRG"/data_sources/exploder_input_38_1000`
Output: `"$rootRG"/data_sources/exploder_output_38_1000`

The purpose of the curation directory is to maintain a working area where the variant haplotype data may be manipulated separately from the reference-data. The curation sub-directories for each locus initially holds soft links to the processed files in

`"$rootRG"/data_sources/GRCH38_sequences_1000/ "$locus"_dir`

Occasionally an edited copy of *`"$locus"`*`_locseq.gb` is used within the curation directory (eg: AK2 for GRCH37) instead of a soft-link.

The input directory is used by the application.

- The input directory requires the presence of a configuration file `config.json`.
- The input sub-directories for each locus hold soft links to files in the curation directories.
- The output directory requires empty folders for each locus present.

The output directory is also used by the application.

- The sub-directories have the same names present in the input directory.
- These are empty to receive the output from the Python GUI

Automated addition of multiple new curation folders

If you are starting from scratch and do **not** already have the directories

`"$rootRG"/GRCH38_sequences_1000_curation,`
or
`"$rootRG"/GRCH37_sequences_1000_curation,`

then:

a) Check, and amend where necessary, the path definitions at the head of
`"$rootRG"/helper_scripts/create_curation_tree.sh`

```
rootapplicationdir="$rootRG"/
rootdatadir=$rootapplicationdir"data_sources/"
targetdir37="GRCH37_sequences_1000"
targetdir38="GRCH38_sequences_1000"
```

b) Execute the script using `37` or *`38`* as a parameter eg:

```
> sh "$rootRG"/helper_scripts/create_curation_tree.sh 38
```

and it will build a set of directories in eg: where `??` is 37 or 38

`"$rootRG"/GRC??_sequences_1000_`**`curation`**, with the same starting names as in

`"$rootRG"/GRCH??_sequences_1000`

It also creates soft links from GRCH38_sequences_1000 as described previously

c) If you *already have* a **curation** directory, and execute the script anyway: you will get many error-reports about files that have been created previously, but these should be harmless.

You should find that any new locus directories will be created within the relevant curation directory.

Automated addition of multiple new input and output folders

If you are starting from scratch and do **not** already have the directories

`"$rootRG"/exploder_input_??_1000` and `"$rootRG"/exploder_output_??_1000` then

a) Check, and amend where necessary, the path definitions at the head of the two scripts **`"$rootRG"/helper_scripts/create_input_directory.sh`**:

```
rootapplicationdir="$rootRG"/
rootdatadir=$rootapplicationdir"data_sources/"
targetdir37="GRCH37_sequences_1000"
input_seq37=$rootdatadir"exploder_input_37_1000"
targetdir38="GRCH38_sequences_1000"
input_seq38=$rootdatadir"exploder_input_38_1000"
```

`"$rootRG"/helper_scripts/create_output_directory.sh`:

```
rootapplicationdir="$rootRG"/
rootdatadir=$rootapplicationdir"data_sources/"
output_seq37=$rootdatadir"exploder_output_37_1000"
input_seq37=$rootdatadir"exploder_input_37_1000"
output_seq38=$rootdatadir"exploder_output_38_1000"
input_seq38=$rootdatadir"exploder_input_38_1000"
```

b) Execute the first script using `37` or *`38`* as a parameter
eg: `> sh "$rootRG"/helper_scripts/create_input_directory.sh 38`

and it will build a set of directories in eg:
`"$rootRG"/exploder_input_38_1000`
with the same names as in `"$rootRG"/GRCH38_sequences_1000`

It **also** creates soft links from GRCH38_sequences_1000_curation as described previously.

Plus it overwrites / creates a soft link
`"$rootRG"/exploder_python/input`
to
`"$rootRG"/exploder_input_38_1000`

c) Execute the second script using `37` or *`38`* as a parameter
eg: `> sh "$rootRG"/helper_scripts/create_output_directory.sh 38`

and it will build a set of empty directories, with the same names as in the output directory, in eg:

`"$rootRG"/exploder_output_38_1000`

Plus it deletes `"$rootRG"/exploder_python/output`

and makes this a soft link to

`"$rootRG"/exploder_output_38_1000`

d) If you *already have* an input and output directory, and execute the script anyway: you will get error-reports for files previously created.

You should find that any **new** locus directories originating from the curation directories will be created within the input and output directories. Just be aware of the possible errors arising from trying to create a soft link that already exists.

Mashup time

This file must exist for `RG_exploder_globals_make.py` to run:
`"$rootRG"/data_sources/exploder_input_38_1000/loci.json`

It is a concatenation of all the individual `"$locus"_transcripts.json` files in the input subdirectories eg:

```
caryodonnell@MacBook-Air data_sources % ls "$rootRG"/data_sources/exploder_input_38_1000/*/*.json
/Users/caryodonnell/Desktop/Replicon//data_sources/exploder_input_38_1000/AK2/AK2_transcripts.json
/Users/caryodonnell/Desktop/Replicon//data_sources/exploder_input_38_1000/ATM/ATM_transcripts.json
...
/Users/caryodonnell/Desktop/Replicon//data_sources/exploder_input_38_1000/STK11/STK11_transcripts.json
/Users/caryodonnell/Desktop/Replicon//data_sources/exploder_input_38_1000/TP53/TP53_transcripts.json
```

To create / renew this file with the latest data:
a) Check, and amend where necessary, the path definitions at the head of the script
`"$rootRG"/helper_scripts/call_mash_json.sh`

```
rootapplicationdir="$rootRG"/
rootdatadir=$rootapplicationdir"data_sources/"
input_seq37=$rootdatadir"exploder_input_37_1000"
input_seq38=$rootdatadir"exploder_input_38_1000"
```

Then execute:

```
> sh "$rootRG"/helper_scripts/call_mash_json.sh 38
```

Finally, as in the instructions at the start of this document:

o) To create the lookup file `"$rootRG"/data_sources/exploder_python/input/config.json`
Run the python script `RG_exploder_globals_make.py` (check values in `set_config_consts`) in the `exploder_python` directory:

```
> cd "$rootRG"/exploder_python/
> python3 RG_exploder_globals_make.py
```

Now the data should be ready for the application, test it out:

p) Finally, run the application
`"$rootRG"/data_sources/exploder_python/RG_exploder_gui.py`

```
> cd "$rootRG"/exploder_python/
> python3 RG_exploder_gui.py
```

Manual maintenance of curation data

This section describes steps that the automated post-processing will do, along with manual maintenance to Variant Haplotype definition files. Managing special cases is also described.

Manual addition of a new curation folder

```
> mkdir "$rootRG"/data_sources/GRCH38_sequences_1000_curation/"$locus"_curation
> cd "$rootRG"/data_sources/GRCH38_sequences_1000_curation/"$locus"_curation
```

Simply soft link to the source data locus directory

```
> ln -s ../ ../GRCH38_sequences_1000/"$locus"_dir
```

Then soft link the following 2 files:

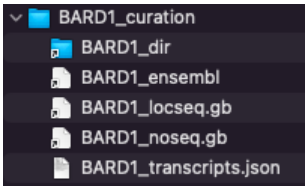
```
> ln -s "$locus"_dir/"$locus"_locseq.gb .
> ln -s "$locus"_dir/"$locus"_noseq.gb .
```

Then **copy** this one:

```
> cp -p "$locus"_dir/"$locus"_transcripts.json .
```

For a simple setup, such as for BARD1, which has no haplotype definitions apart from the reference, nothing more needs to be done.

```
BARD1_dir -> ../../GRCH38_sequences_1000/BARD1_dir/
BARD1_ensembl -> BARD1_dir/BARD1_ensembl
BARD1_locseq.gb -> BARD1_dir/BARD1_locseq.gb
BARD1_noseq.gb -> BARD1_dir/BARD1_noseq.gb
BARD1_transcripts.json
```



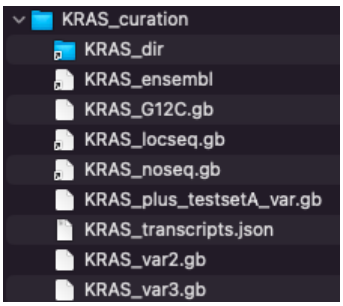
A link to `BARD1_ensembl` is not required in this directory, but its presence can be useful.

Other loci are described below; chosen to demonstrate both the standard and less-obvious maintenance options available.

For AK2

```
> cd "$rootRG"/data_sources/GRCH38_sequences_1000_curation/AK2_curation
> ls -l
```

```
AK2_EB0001.gb
AK2_EB0002.gb
AK2_EB0003.gb
AK2_EB0004.gb
AK2_EB0005.gb
AK2_EB0006.gb
AK2_dir -> ../../GRCH38_sequences_1000/AK2_dir/
AK2_ensembl -> AK2_dir/AK2_ensembl
AK2_locseq.gb -> AK2_dir/AK2_locseq.gb
AK2_noseq.gb -> AK2_dir/AK2_noseq.gb
AK2_transcripts.json
```



Each of the haplotype definition files, in **bold**, contain these essential components:

A) The Header section, note the **0 bp** definition, as there is no sequence in the file:

```
LOCUS       1                         0 bp    DNA             HTG 19-AUG-2022
DEFINITION  Homo sapiens chromosome 1 GRCh38 partial sequence 33006986..33081996
            reannotated via EnsEMBL.

ACCESSION   chromosome:GRCh38:1:33006986:33081996:1
VERSION     chromosome:GRCh38:1:33006986:33081996:1
FEATURES             Location/Qualifiers
     source          1..75011
                     /organism="Homo sapiens"
                     /db_xref="taxon:9606"
     gene            complement(1001..74011)
                     /gene="ENSG00000004455.18"
                     /locus_tag="AK2"
                     /note="adenylate kinase 2 [Source:HGNC
                     Symbol;Acc:HGNC:362]"
```

This section should be the same as found in the file `AK2_noseq.gb` which can be used as a template for their creation.

B) A definition of the variation(s) in the Feature table, eg:

`AK2_EB0001.gb` has just one:

```
     variation       7582..7583
                      /replace="GG/G"
```

`AK2_EB0001.gb` also has one:

```
      variation       14469..14471
                      /replace="TCA/-"
```

C) To create a file defining a new variant, options include:

1) Use a SNP or other identifier and look in the file `AK2_ensembl` (the uncompressed, unfiltered source file) or in `AK2_dir/AK2_filtervar.gb`

eg: dbSNP:**rs1553151177** (the `AK2_EB0001` variant) can be found alongside other definitions

```
      variation       7582..7582
                      /replace="G/T"
                      /db_xref="dbSNP:rs1241229733"
      variation       7582..7582
                      /replace="HGMD_MUTATION"
                      /db_xref="HGMD-PUBLIC:CD090014"
      variation       7582..7583
                      /replace="GG/G"
                      /db_xref="dbSNP:rs1553151177"
```

Edit the required lines into a copy of `AK2_noseq.gb`
Just be certain NOT to include overlapping definitions; the application does not recognise these overlaps and is likely to give incorrect output.

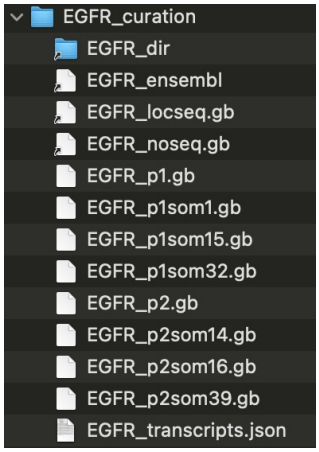
2) You may be able to edit a new definition from other markers you recognise, or using offsets. This manual method is very error-prone.

3) To create a *validated* position for a sequence-modification: first complete all the other "adding a new locus" steps; then run the GUI and use the "Create a new *locus* Haplotype Variant" feature, along with "Save Source Features". That will generate an output file with a name you supply that can be copied straight back into this directory. Worked example for EGFR below.

For EGFR

```
> cd "$rootRG"/data_sources/GRCH38_sequences_1000_curation/EGFR_curation
```

```
caryodonnell@MacBook-Air EGFR_curation % pwd
/Users/caryodonnell/Desktop/Replicon/data_sources/
exploder_input_38_1000/EGFR/EGFR_curation
caryodonnell@MacBook-Air EGFR_curation % ls -l
total 80
EGFR_dir -> ../../GRCH38_sequences_1000/EGFR_dir/
EGFR_ensembl -> EGFR_dir/EGFR_ensembl
EGFR_locseq.gb -> EGFR_dir/EGFR_locseq.gb
EGFR_noseq.gb -> EGFR_dir/EGFR_noseq.gb
EGFR_p1.gb
EGFR_p1som1.gb
EGFR_p1som15.gb
EGFR_p1som32.gb
EGFR_p2.gb
EGFR_p2som14.gb
EGFR_p2som16.gb
EGFR_p2som39.gb
EGFR_transcripts.json
```



The variant definitions here are hierarchical, it uses two different haplotype definitions, each extracted originally from `EGFR_ensembl`, with extra commentary added from other sources:

`EGFR_p1.gb` **(for "parent 1")**

```
     variation       156018..156018
                     /replace="G/A"
                     /db_xref="dbSNP:rs55959834"
                     /consequence="dbSNP:synonymous_variant,genic_downstream_transcript_variant"
                     /consequence="dbSNP:coding_sequence_variant"
                     /comment="ensembl:minor allele exon18, synonymous variant at v low freq 0.001"

     variation       160511..160511
                     /replace="G/A"
                     /db_xref="dbSNP:rs62457092"
                     /consequence="dbSNP:genic_downstream_transcript_variant,intron_variant"
                     /comment="ensembl:intron_variant 19_20 minor allele at 0.32"
```

`EGFR_p2.gb` **(for "parent 2")**

```
     variation       159798..159798
                     /replace="A/G"
                     /db_xref="dbSNP:rs845552"
                     /consequence="dbSNP:intron_variant,genic_downstream_transcript_variant"
                     /comment="ensembl:intron_variant 19_20 minor allele at 0.45"

     variation       163354..163354
                     /replace="G/A"
                     /db_xref="dbSNP:rs1050171"
                     /consequence="dbSNP:genic_downstream_transcript_variant,synonymous_variant"
                     /consequence="dbSNP:missense_variant,non_coding_transcript_variant"
                     /consequence="dbSNP:coding_sequence_variant"
                     /comment="ensembl:minor allele exon 20, synonymous variant at 0.43"
```

The other files have further variants added onto these basic haplotypes

IMPORTANT:

The header sections of the 'parent1' files must agree with the source of the variant.

EGFR_p1.gb header:

```
LOCUS       7 0 bp DNA HTG 27-FEB-2022
DEFINITION  Homo sapiens chromosome 7 GRCh38 partial sequence 55018017..55212628 reannotated
            via EnsEMBL
ACCESSION   chromosome:GRCh38:7:55018017:55212628:1
VERSION     chromosome:GRCh38:7:55018017:55212628:1
COMMENT     /consequence and /comment annotation by Replicon Genetics from public domain sources
FEATURES             Location/Qualifiers
     source          1..194612
                     /organism="Homo sapiens"
                     /db_xref="taxon:9606"
     gene            1001..193612
                     /gene=ENSG00000146648.20
                     /locus_tag="EGFR"
```

EGFR_p1som1.gb: **(som1 for "somatic 1")**

```
LOCUS       7 0 bp DNA HTG 27-FEB-2022
DEFINITION  Homo sapiens chromosome 7 GRCh38 partial sequence 55018017..55212628 reannotated
            via EnsEMBL
ACCESSION   chromosome:GRCh38:7:55018017:55212628:1
VERSION     chromosome:GRCh38:7:55018017:55212628:1
COMMENT     /consequence and /comment annotation by Replicon Genetics from public domain sources
FEATURES             Location/Qualifiers
     source          1..194612
                     /organism="Homo sapiens"
                     /db_xref="taxon:9606"
     gene            1001..193612
                     /gene=ENSG00000146648.20
                     /locus_tag="EGFR"
```

Intended capability not currently available here crossed out:

~~They DO NOT need to be *exactly the same* as the header in the reference-sequence file~~

~~**EGFR_locseq.gb**:~~

~~LOCUS 7 194612 bp DNA HTG 27-FEB-2022~~
~~DEFINITION Homo sapiens chromosome 7 GRCh38 partial sequence 55018017..55212628 reannotated via EnsEMBL~~
~~ACCESSION chromosome:GRCh38:7:55018017:5521**2628**:1~~
~~VERSION chromosome:GRCh38:7:55018017:55212628:1~~
~~FEATURES Location/Qualifiers~~
~~source 1..194612~~
~~/organism="Homo sapiens"~~
~~/db_xref="taxon:9606"~~
~~gene 1001..193612~~
~~/gene=ENSG00000146648.20 ← The difference is gene-id version is acceptable~~
~~/locus_tag="EGFR"~~

~~The range for the reference definition is **GRCh38:7:55018017:55212628:1**~~

~~The range for the variant definition is **GRCh38:7:55018517:55212128:1**~~

- ~~Both the GRCh version and strand are the same: this is essential~~
- ~~The sequence range of the variant is wholly contained within the reference range~~
  - ~~If not, the application will generate a warning message and should ignore the haplotype completely.~~
  - ~~The application detects the disparity and recalculates the offset~~

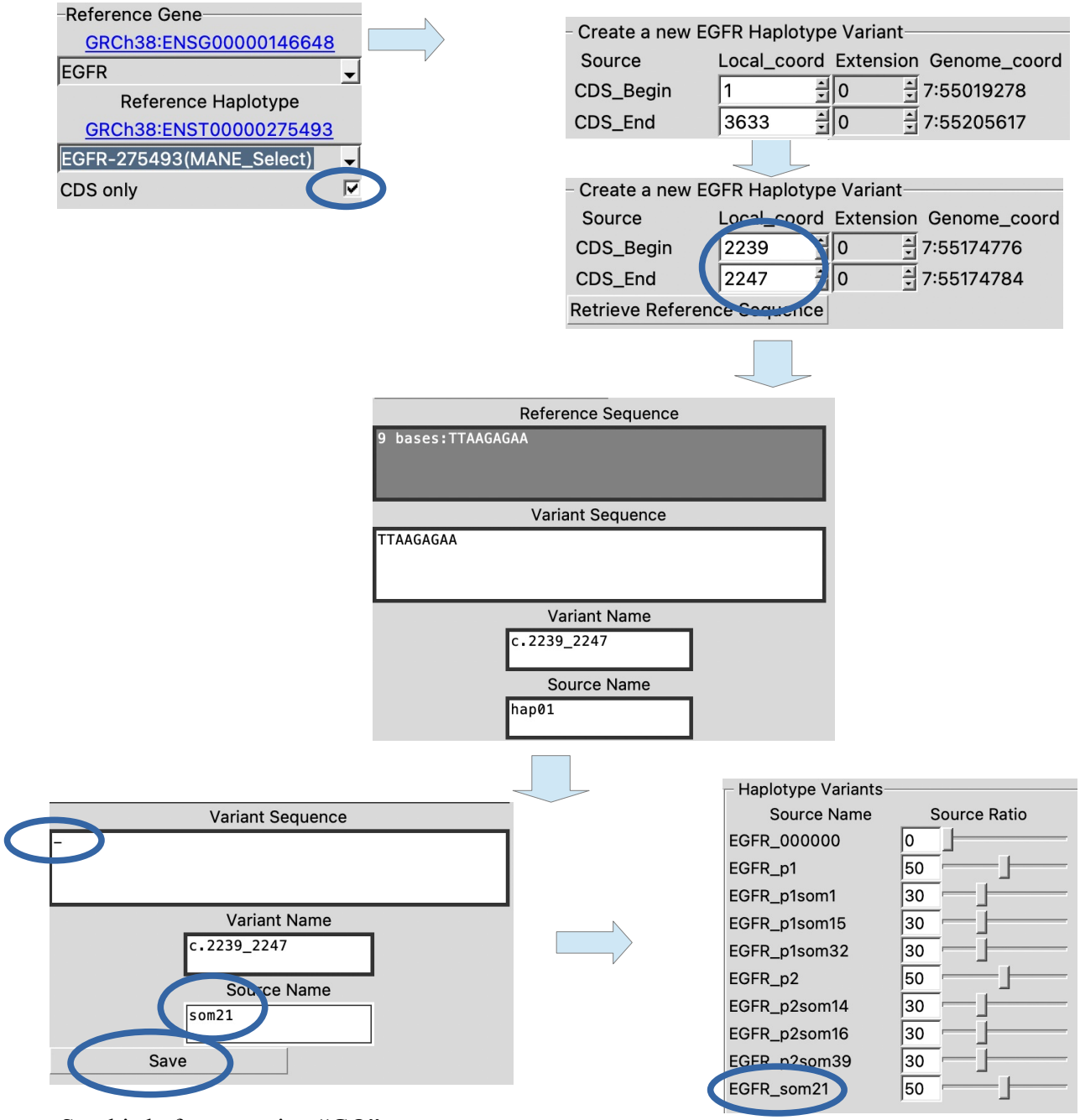
~~In this way, variant definition files do **not** need to be regenerated each time the source files are updated, and where only a small redefinition of the gene range takes place.~~

The inspiration, and naming, for this set of EGFR variants comes from table 2 of *"Molecular characteristics and clinical outcomes of EGFR exon 19 indel subtypes to EGFR TKIs in NSCLC patients"* by Su et al Oncotargetv.8(67); 2017 Dec 19

Subtypes are defined at CDS like this: Subtype 21 - **`c.2239_2247del9`**

Additional variants:
Here's how to add two of the subtypes described using the application GUI rather than alternative, painstaking methods.

Subtype 21

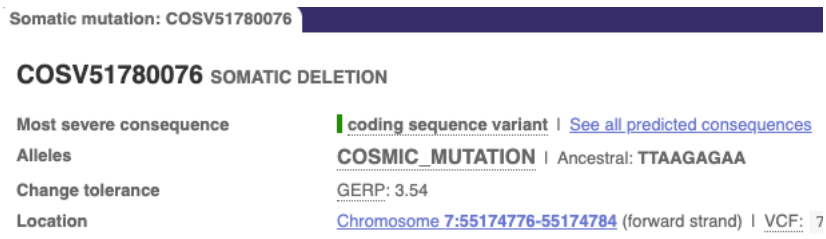


Set this before pressing "GO"

In the output will be a file name like **EGFR-275493-CDS_paired_DNASeq_som21.gbout**

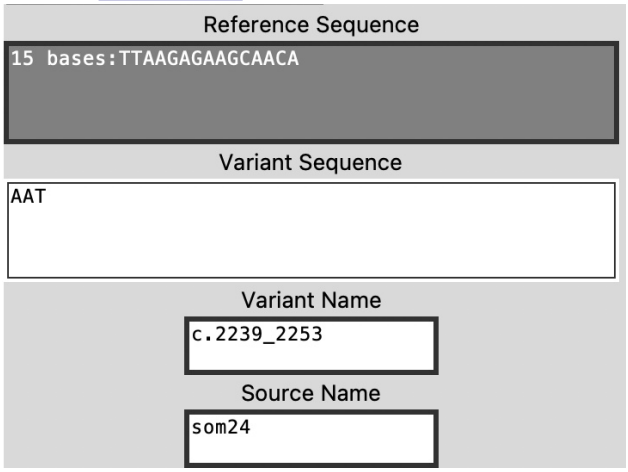
```
LOCUS       7                          0 bp    DNA              HTG 27-FEB-2022
DEFINITION  Homo sapiens chromosome 7 GRCh38 partial sequence 55018017..55212628
            reannotated via EnsEMBL.
ACCESSION   chromosome:GRCh38:7:55018017:55212628:1
VERSION     chromosome:GRCh38:7:55018017:55212628:1
KEYWORDS    .
SOURCE      .
  ORGANISM  .
            .
FEATURES             Location/Qualifiers
     source          1..194612
                     /organism="Homo sapiens"
                     /db_xref="taxon:9606"
     gene            1001..193612
                     /gene="ENSG00000146648.20"
                     /locus_tag="EGFR"
                     /note="epidermal growth factor receptor [Source:HGNC
                     Symbol;Acc:HGNC:3236]"
     variation       156760..156768
                     /replace="TTAAGAGAA/-"
                     /db_xref="som21_1:c.2239_2247"
                     /global_range="GRCh38:7:55174776:55174784:1"
ORIGIN
```

Note that the location calculated by the application matches the assigned COSM6218 entry

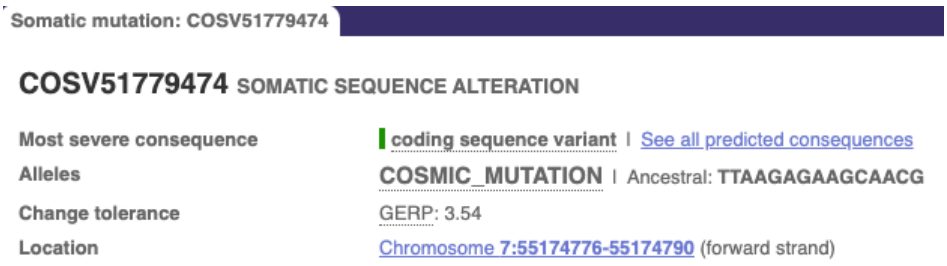


NB: The header matches that of the reference **EGFR_locseq.gb**

Subtype 24 - `c.2239_2253>aat` COSM51503




```
variation       156760..156774
                /replace="TTAAGAGAAGCAACA/AAT"
                /db_xref="som24_1:c.2239_2253"
                /global_range="GRCh38:7:55174776:55174790:1"
```



A preferable method would be to use VCF files, but the application is not currently set up for this.

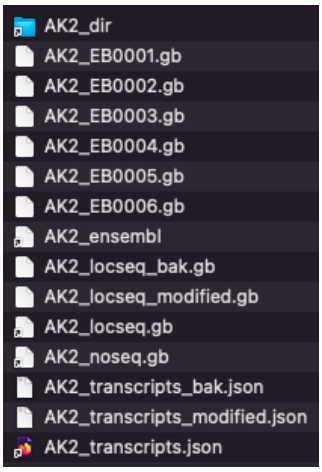
For AK2 in GRCh37

The MANE_Select transcript for AK2 is not defined in GRCh37, but exists in GRCh38 with the identifier `ENST00000672715`

After manually identifying the differing offset-positions between GRCh37 and GRCh38 for exons shared with other transcripts, it was possible to create mRNA and CDS join features for the missing features in GRCh37. To verify this, the transcripts created from each version by the application were aligned. The sequence for this transcript differs by a single base substitution at one location between the two genome builds.

```
AK2_locseq.gb -> AK2_locseq_modified.gb
AK2_locseq_bak.gb
AK2_locseq_modified.gb
AK2_noseq.gb -> AK2_dir/AK2_noseq.gb
AK2_transcripts.json -> AK2_transcripts_modified.json
AK2_transcripts_bak.json
AK2_transcripts_modified.json
```

AK2_locseq_bak.gb is the original version of AK2_locseq.gb



The new file is maintained as `AK2_locseq_modified.gb` with `AK2_locseq.gb` now a soft-link to it and the original saved as `AK2_locseq_bak.gb`

Comparing the feature tables between the two builds:

`AK2_locseq_modified.gb for` GRCh37
Note the modified transcript identifier: from `"ENST00000672715.1"` to `"ENST00000672715m.1"`

The 'm' modifier is recognised in the GUI to generate a URL to link to Ensembl GRCh38 instead of GRCh37.

```
LOCUS       1 75013 bp DNA HTG 12-AUG-2022
DEFINITION  Homo sapiens chromosome 1 GRCh37 partial sequence 33472585..33547597 reannotated
            via EnsEMBL
ACCESSION   chromosome:GRCh37:1:33472585:33547597:1
VERSION     chromosome:GRCh37:1:33472585:33547597:1
FEATURES             Location/Qualifiers
     source          1..75013
                     /organism="Homo sapiens"
                     /db_xref="taxon:9606"
     gene            complement(1001..74013)
                     /gene=ENSG00000004455.12
                     /locus_tag="AK2"
                     /note="adenylate kinase 2 [Source:HGNC Symbol;Acc:362]"
     mRNA            join(complement(29753..29887),complement(17459..17584),
                     complement(14610..14720),complement(14384..14478),
                     complement(7539..7611),complement(1003..6419))
                     /gene="ENSG00000004455.12"
                     /standard_name="ENST00000672715m.1"
                     /comment="RG:copied from GRCH38 as not present in 37 download"
     CDS             join(complement(29753..29845),complement(17459..17584),
                     complement(14610..14720),complement(14384..14478),
                     complement(7539..7611),complement(6198..6419))
                     /gene="ENSG00000004455.12"
                     /protein_id="ENSP00000499935.1"
                     /note="transcript_id=ENST00000672715m.1"
```

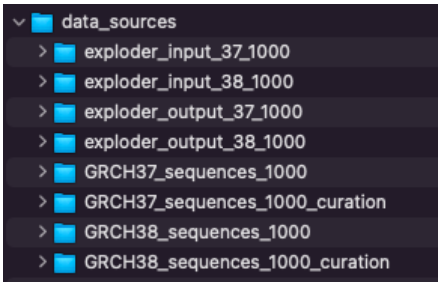
AK2_locseq.gb for GRCh38:

```
LOCUS       1 75011 bp DNA HTG 19-AUG-2022
DEFINITION  Homo sapiens chromosome 1 GRCh38 partial sequence 33006986..33081996 reannotated
            via EnsEMBL
ACCESSION   chromosome:GRCh38:1:33006986:33081996:1
VERSION     chromosome:GRCh38:1:33006986:33081996:1
FEATURES             Location/Qualifiers
     source          1..75011
                     /organism="Homo sapiens"
                     /db_xref="taxon:9606"
     gene            complement(1001..74011)
                     /gene=ENSG00000004455.18
                     /locus_tag="AK2"
                     /note="adenylate kinase 2 [Source:HGNC
                     Symbol;Acc:HGNC:362]"
…
     mRNA            join(complement(29751..29898),complement(17457..17582),
                     complement(14608..14718),complement(14382..14476),
                     complement(7537..7609),complement(1001..6417))
                     /gene="ENSG00000004455.18"
                     /standard_name="ENST00000672715.1"
     CDS             join(complement(29751..29843),complement(17457..17582),
                     complement(14608..14718),complement(14382..14476),
                     complement(7537..7609),complement(6196..6417))
                     /gene="ENSG00000004455.18"
                     /protein_id="ENSP00000499935.1"
                     /note="transcript_id=ENST00000672715.1"
```

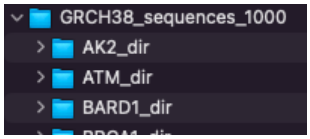
The file `AK2_transcripts.json` has been modified to incorporate the additional, modified transcript identifier. The original, modified and soft-linked files are managed in the same manner as for `AK2_locseq.gb`.

Directory management

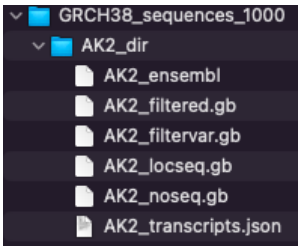
A) Screen shots from MacOS Finder



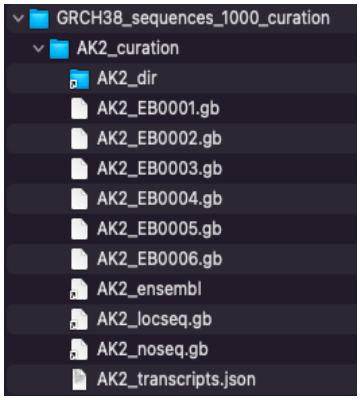
At the root level are these directories



GRCH38_sequences_1000/
locus**_dir** initially hold the downloaded **ensembl.txt.gz** files of each locus

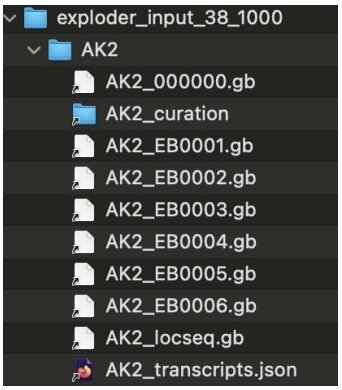


After automated processing of **ensembl.txt.gz**, these files remain in each *locus*_**dir** sub directory:
*locus*_**ensembl** – the original, now unzipped, downloaded file
*locus*_**filtered.gb** – same as *locus*_**ensembl**, minus unwanted annotation
*locus*_**filtervar.gb** – as above, but without mRNA & CDS features, without sequence ; just the variations
*locus*_**locseq.gb** – everything in *locus*_**filtered.gb**, without the variations; this is the Reference Source used in the application.
*locus*_**noseq.gb –** A no-sequence, no-features version. This is used as the input for *locus*_00000 haplotype in the application. It is a blank template for curating other haplotypes. Find known variants in *locus*_**locseq.gb**
*locus*_**transcripts.json** – lookup information for the application**.** These are later concatenated into config.json



A curation directory is used to demarcate the automated-processed files from the hand-created haplotye definitions.
Here: AK2_EB0001.gb … AK2_EB0006.gb

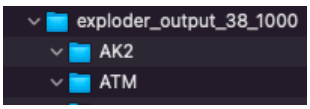
Other files in this *locus*_**curation** directory are usually soft linked to the above. There may be exceptions to this, notably in **GRCH37_sequences_1000_curation** - the original ***AK2*_locseq.gb** and ***AK2*_transcripts.json** have been hand-edited to include the MANE_Select transcript that is absent in GRCH37, but is defined in GRCH37.



exploder-input_38_1000
The **exploder_python/input** directory is a soft link to here

Where all files in
exploder_input_38_1000/*locus*
are soft links to files in
GRCH38_sequences_1000_curation/*locus*_**curation**
This allows for hand-culling or renaming of links in the input directory whilst retaining them in the curation directory or under a different name.
Eg: compare the EGFR directories.

Finally – **exploder-output_38_1000**
Where the sub-directories for each locus are empty to receive the output from the application.
The **exploder_python/output** directory is a soft link to here



Recorded information on loci

NB: positions & annotation seems to change with Ensembl release versions. There may be inconsistency in the *numbers* stated within the URL (r=x:*nnn-nnn*) for the Ensembl ID and the genomic *location* this actually links to, but the URL seems to correct itself over time.

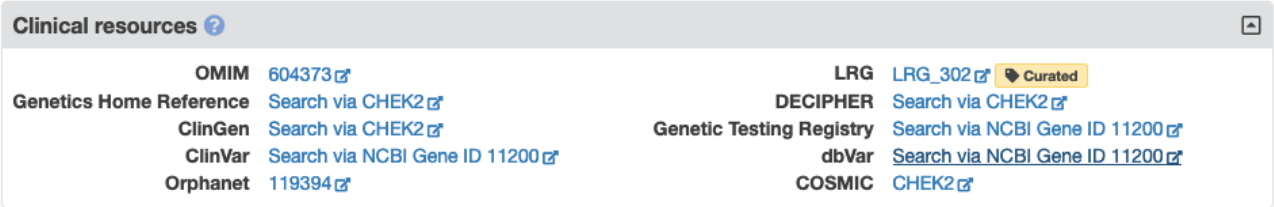
Note that it's normal for Ensembl and NCBI to show different position mappings; typically small numerical-differences at the start and end of a locus. All below are Ensembl mappings

Ensembl Release 96 Apr 2019, GRCh38.p12 – GENCODE release 30

OMIM	HGNC symbol report	Ensembl ID	Genomic Location GRCh38.p12	Publication	NIH genetics home reference
ATM	HGNC:795	ENSG00000149311	11: 108,219,552-108,372,034 (+)		ATM serine/threonine kinase ataxia telangiectasia
BARD1	HGNC:952	ENSG00000138376	2: 214,723,966-214,811,363 (-)		BRCA1 associated RING domain 1
BRCA1	HGNC:1100	ENSG00000012048	17: 43,041,776-43,172,764 (-)		DNA repair associated
BRCA2	HGNC:1101	ENSG00000139618	13: 32,313,779-32,401,961 (+)		DNA repair associated
BRIP1	HGNC:20473	ENSG00000136492	17: 61,677,621-61,867,166 (-)		BRCA1 interacting protein C-terminal helicase 1
CDH1	HGNC:1748	ENSG00000039068	16: 68,735,328-68,837,505 (+)		cadherin 1
CDK12	HGNC:24224	ENSG00000167258	17: 39,459,444-39,566,974 (+)		cyclin dependent kinase 12
CHEK1	HGNC:1925	ENSG00000149554	11: 125,624,114-125,677,277 (+)		checkpoint kinase 1
CHEK2	HGNC:16627	ENSG00000183765	22: 28,686,650-28,743,515 (-)	Breast Cancer (Dove Med Press). 2017; 9: 331–335.	checkpoint kinase 2
EPCAM	HGNC:11529	ENSG00000119888	2: 58,157,601-58,243,014 (+)		epithelial cell adhesion molecule
FANCL	HGNC:20748	ENSG00000115392	2: 58,159,243-58,241,372 (-)		FA complementation group L
KRAS	HGNC:6407	ENSG00000133703	12: 25,203,867-25,251,858 (-)		KRAS proto-oncogene, GTPase
MLH1	HGNC:7127	ENSG00000076242	3: 36,992,181-37,052,069 (+)		mutL homolog 1 - DNA repair
MSH2	HGNC:7325	ENSG00000095002	2: 47,397,766-47,668,349 (+)		mutS homolog 2
MSH6	HGNC:7329	ENSG00000116062	2: 47,693,239-47,812,392 (+)		mutS homolog 6
NBN	HGNC:7652	ENSG00000104320	8: 89,931,939-90,004,625 (-)		nibrin
NF1	HGNC:7765	ENSG00000196712	17: 31,089,184-31,387,859 (+)		Neurofibromin 1
PALB2	HGNC:26144	ENSG00000083093	16: 23,602,397-23,642,073 (-)		partner and localizer of BRCA2
PMS2	HGNC:9122	ENSG00000122512	7: 5,970,162-6,009,869 (-)		PMS1 homolog 2, mismatch repair system component
PPP2R2A	HGNC:9304	ENSG00000221914	8: 26,289,885-26,374,303 (+)		protein phosphatase 2 regulatory subunit Balpha
PTEN	HGNC:9588	ENSG00000171862	10: 87,861,459-87,974,096 (+)		phosphatase and tensin homolog
PTEN_a	HGNC:9588	ENSG00000284792	CHR_HG2334_PATCH: 87,861,382-87,968,399 (+)		phosphatase and tensin homolog (alternative mapping)
RAD51B	HGNC:9822	ENSG00000182185	14: 67,801,571-68,748,426(+)		RAD51 paralog B
RAD51C	HGNC:9820	ENSG00000108384	17: 58,691,713-58,736,471 (+)		RAD51 paralog C
RAD51D	HGNC:9823	ENSG00000185379	17: 35,091,622-35,122,108 (-)		RAD51 paralog D

RAD54L	HGNC:9826	ENSG00000085999	1: 46,247,073-46,279,088 (+)		RAD54 like
STK11	HGNC:11389	ENSG00000118046	19: 1,176,541-1,229,452(+)		serine/threonine kinase 11
TP53	HGNC:11998	ENSG00000141510	17: 7,661,264-7,688,065 (-)		tumor protein p53

For clinical evidence on variations, from HGNC symbol report: click on Clinical Resources/Clinvar
For locations: Clinical Resources/Genetic Testing Registry



NB: Orphanet link has “diagnostic tests” link that lists laboratories testing for this gene.

Alternative possibilities:

a) Use Entrez API as an alternative to a manual selection: direct extraction of locus into application?
Entrez API requires a specific account, but is an obvious development

b) Ensembl Biomart
Do online search: tutorials etc, but beware broken links and not-working notices

Use Biomart to download only non-synonymous SNPs and indels?
Eg: for ATM

```
http://www.ensembl.org/biomart/martview/6f94c204c7331054b28277e0ed89d23c?
VIRTUALSCHEMANAME=default&ATTRIBUTES=hsapiens_snp.default.snp.refsnp_id|
hsapiens_snp.default.snp.refsnp_source|hsapiens_snp.default.snp.chr_name|
hsapiens_snp.default.snp.chrom_start|hsapiens_snp.default.snp.chrom_end|
hsapiens_snp.default.snp.consequence_type_tv|
hsapiens_gene_ensembl.default.snp.ensembl_gene_id_version&FILTERS=hsapiens_snp.d
efault.filters.chr_name."11"|
hsapiens_snp.default.filters.so_mini_parent_name."nonsynonymous_variant"&VISIBLE
PANEL=resultspanel


Your reference is martquery_0704132240_850.txt.gz.
```

Modifications log

Date	Section	Changes/reasons
13th February 2025	Multiple sections	Updating screenshots and superseding script-usage instructions
7th February 2025	"Automated data processing for downloaded Ensembl data"	Documenting additional processing Python script, which supersedes previous.
28th January 2025		Corrections
27th January 2025	New Sections: "Downloading a sequence file for a locus from NCBI" "Data processing for downloaded NCBI data"	Section similar to that for existing Ensembl download, re-written for NCBI-derived datas
6th May 2023	First version	