PROYECTO FINAL SQL

Máster en Bioinformática Aplicada a la Medicina Personalizada y la Salud

Alumno: Mar Batlle

Fecha: Enero 2020

PARTE 1

1.0. Variant Summaries

Con el objetivo de desarrollar los programas descritos en el enunciado, usaremos el programa presentado en clase; *clinvar_parser*, para crear las bases de datos de SQL con el contenido extraído mediante las release de *variant_summary.txt*, una con el release congelado de Junio y otra de Diciembre de 2020, creando así; *clinvar_jun.db* y *clinvar_dec.db*, respectivamente, como se observa a continuación.

```
python3 clinvar_parser.py clinvar_jun.db variant_summary_2020-06.txt.gz   #
Release variant_summary Junio
python3 clinvar_parser.py clinvar_dec.db variant_summary_2020-12.txt.gz   #
Release variant_summary Diciembre
```

Al realizar este comando, parece que todos los datos se han cargado correctamente en la database correspondiente a la release del mes de junio, pero, al hacer el procedimiento respectivo al mes de diciembre, en las columnas donde se tendrían que indicar los alelos de referencia y los alterados, aparece solamente valores na. Para determinar que podía estar causando este problema, se ha consultado el README de Clinvar, donde se ha visto estas notas:

```
Please note: Beginning in August 2020, the values for referenceAllele and
alterateAllele were reverted to represent right-shifted locations. Columns
for vcf_pos, vcf_ref, and vcf_alt were added to represent the left-shifted
location, as in the VCF standard. These locations are reported so that they
are consistent with the data in the XML files. In each row, either
vcf_pos/vcf_ref/vcf_alt OR start/stop/referenceAllele/alternateAllele are
provided, but not both.
```

Al leer esta nota, se puede determinar que ha habido un cambio en el formato de Clinvar, por lo tanto, para realizar este proyecto sera necesario cambiar el programa original; *clinvar_parser*. El procedimiento que vamos a seguir es crear una copia de el programa original para leer los releases posteriores al mes de agosto, que tendrá por nombre *clinvar_parser_new*, pero con los cambios mínimos para poder leer este nuevo formato. De esta manera, en el nuevo programa se ha modificado ref_allele para que lea los valores de la columna ReferenceAlleleVCF, alt_allele para que la los de AlternateAlleleVCF y se ha añadido una nueva columna, vcf_pos.

Una vez introducidos todos los cambios necesarios a *clinvar_parser_new.py*, se ha creado la tabla de datos SQL correspondiente.

```
python3 clinvar_parser_new.py clinvar_dec.db variant_summary_2020-12.txt.gz
# Release variant_summary Diciembre
```

1.A. Artículos científicos de ClinVar

El objetivo de este problema es crear un programa similar a clinvar_parser.py pero que se encargue de cargar información de los artículos científicos relacionados con las variantes dentro de la base de datos pública Clinvar. Para saber que fichero procesar, se ha analizado el documento facilitado; *tab_delimited/README*, donde se indica que informes contiene el conjunto de datos de ClinVar. Al analizar este documento, se ha visto que en este caso, la información requerida para completar este programa está contenida en el subdirectorio 4; *var_citations.txt*, que como se observa a continuación contiene información de la fuente de la citación y su ID, junto a otra información representativa de la variable:

```
4. var_citations.txt
--------------------
Generated weekly
Not archived

A tab-delimited report of citations associated with data in ClinVar,
connected to the AlleleID, the VariationID, and either rs# from dbSNP or
nsv in dbVar.

AlleleID                        integer value as stored in the AlleleID
field in ClinVar  (//Measure/@ID in the XML)
VariationID                     The identifier ClinVar uses to anchor its
default display. (in the XML,  //MeasureSet/@ID)
rs                                        rs identifier from dbSNP, null if
missing
nsv                                  nsv identifier from dbVar, null if
missing
citation_source                   The source of the citation, either PubMed,
PubMedCentral, or the NCBI Bookshelf
citation_id                          The identifier used by that source
```

El archivo *var_citations.txt* se genera semanalmente y no es archivado, por lo tanto, se usara al largo de este proyecto el archivo con fecha de 19-12-2020. Para obtener localmente esta información, se ha descargado el archivo en nuestra carpeta de trabajo. El documento descargado, como se ha comprobado a continuación, contiene todas las columnas comentadas anteriormente y presenta 925468 líneas de datos (incluyendo el header).

```
$ cat var_citations.txt | wc -l         # Número líneas de datos
  925468
$ cat var_citations.txt | head -n 1     # Nombres de las columnas
```

```
  #AlleleID       VariationID     rs      nsv     citation_source
citation_id
```

Para cargar los datos de este documento en una base de datos SQL será necesario crear un programa mediante python. El primer paso es la creación de una tabla para guardar estos datos, que llamaremos *var_citations*, junto a la descripción de las columnas que la forman. En este caso, las que se han considerado esenciales para crear una tabla representativa, son: *allele_id*, *variation_id*, *citation_source* y *citation_id*. También es interesante mencionar que al considerar que toda esta información es esencial para el objetivo de este programa, se han impuesto restricciones fuertes, ignorando los valores nulos. Al analizar los datos para determinar que columna es la más adecuada para ser usada como *PRIMARY KEY*, se ha visto que las presentes presentavan valores repetidos, por lo que se ha decidido usar una autoincremental, *entry_id*, para cumplir esta función.

```
CREATE TABLE IF NOT EXISTS var_citations (
    entry_id INTEGER PRIMARY KEY AUTOINCREMENT,
    allele_id INTEGER NOT NULL,
    variation_id INTEGER NOT NULL,
    citation_source VARCHAR(64) NOT NULL,
    citation_id VARCHAR(64) NOT NULL
)
```

Este paso de creación de la tabla *var_citations*, ha sido incluido en el script del programa, *clinvar_citations.py*, añadiendo IF NOT EXIST, para así, poder ejecutarlo, exista o no las tabla. Este programa, se ha creado usando el script trabajado en clase, *clinvar_parser.py* como base y guía, pero adaptandolo para ajustarse a los requerimientos de este, teniendo en cuenta que este programa debe ser capaz de crear su propia tabla dentro de las bases de datos generadas al cargar una release en concreto de ClinVar usando *clinvar_parser.py*, para así poder completar la segunda parte del proyecto.

En este script se han realizado otras modificaciones, se ha optado por ejemplo, por no usar la librería gzip para abrir el fichero tabular, ya que en este caso, el archivo descargado de ClinVar no se presenta comprimido. Dentro del script de python, se ha determinado las columnas que deseamos introducir, junto a las adecuadiones de su nombre, para finalizar la edición del script.

```
else:
    columnValues = re.split(r"\t",wline)

    for iCol, vCol in enumerate(columnValues):
        if len(vCol) == 0 or vCol == "-":
            columnValues[iCol] = None

    allele_id = int(columnValues[headerMapping["AlleleID"]])
    variation_id = int(columnValues[headerMapping["VariationID"]])
    citation_source = columnValues[headerMapping["citation_source"]]
    citation_id = columnValues[headerMapping["citation_id"]]

    cur.execute("""
        INSERT INTO var_citations(
```

```
            allele_id,
            variation_id,
            citation_source,
            citation_id)
        VALUES(?,?,?,?)
    """, (allele_id,variation_id,citation_source, citation_id,))

    # The autoincremented value is got here
    entry_id = cur.lastrowid
```

Posteriormente, se ha ejecutado el programa creado, *clinvar_citations.py*, añadiendo así la tabla *var_citations* a los ficheros de datos creados anteriormente. Para comprobar que se han leído y cargado todas las líneas, se ha usado SQLITE3 para contar las líneas de datos de la tabla var_citations, que como se observa, es el mismo valor (excluyendo el header), que el observado anteriormente.

```
# DB Junio
python3 clinvar_citations.py clinvar_jun.db var_citations.txt
sqlite3 clinvar_jun.db
    -- SQLite
    SELECT COUNT(*)
    FROM var_citations;
    925467

# DB Diciembre
python3 clinvar_citations.py clinvar_dec.db var_citations.txt
sqlite3 clinvar_dec.db
    -- SQLite
    SELECT COUNT(*)
    FROM var_citations;
    925467
```

1.B. Estadísticas de variantes por gen

El objetivo de este segundo problema es crear un programa capaz de cargar información de las estadísticas de variantes por gen. Para decidir que fichero procesar, se ha vuelto a analizar el README de ClinVar, y en este caso, se ha visto que la información requerida se encuentra en el subdirectorio 1; gene_specific_summary.txt, que como se observa a continuación contiene información sobre las estadísticas de variables por gen:

```
1. gene_specific_summary
------------------------
Generated weekly
Archived monthly ( first Thurday of each month)

Although this report is generated each week, it is currently based on
statistics that are captured the first day of each month.
Therefore there will be some discrepancies between what is reported in this
file and what may be viewed interactively on the web.
```

```
A tab-delimited report, for each gene, of the number of submissions and the
number of different variants (alleles).
Because some variant-gene relationships are submitted, and some are
calculated from overlapping annotation, in January of 2015, the report was
modified to indicate when the gene-variant relationship was submitted.

Symbol                 Gene symbol (if officially named, from HGNC, else
from NCBI's Gene database)
GeneID                 Unique identifier from  NCBI's Gene database
Total_submissions         Total submissions to ClinVar with variants
in/overlapping this gene
Total_alleles          Number of alleles submitted to ClinVar for this gene
Submissions_reporting_this_gene
                       Subset of the total submissions that also reported the
gene
Alleles_reported_Pathogenic_Likely_pathogenic
                       Number of variants reported as pathogenic or likely
pathogenic
                       Excludes structural variants that may overlap a gene
Gene_MIM_Number           The MIM number for this gene
Number_Uncertain       Submissions with an interpretation of 'Uncertain
significance'
Number_with_conflicts  Number of VariationIDs for this gene with
conflicting interpretations
```

Este archivo, al contrario del usado en el apartado anterior, si que se archiva mensualmente, por lo tanto, se usara el archivo con fecha de 04-06-2020 y 03-12-2020 respectivamente. Para obtener localmente esta información, se ha descargado los archivos comprimidos a nuestra carpeta de trabajo. Los documentos descargados, como se comprueba a continuación, contienen todas las columnas comentadas anteriormente y presentan 32955 y 32985 líneas de datos, incluyendo dos headers, respectivamente.

```
# Fichero Junio
$ gunzip -c gene_specific_summary_2020-06.txt.gz | head -n 2  # Nombres de
las columnas
    #Overview of data in ClinVar by gene, dated June 2, 2020
    #Symbol GeneID  Total_submissions       Total_alleles
Submissions_reporting_this_gene
Alleles_reported_Pathogenic_Likely_pathogenic   Gene_MIM_number
Number_uncertain        Number_with_conflicts

# Fichero Diciembre
$ gunzip -c gene_specific_summary_2020-12.txt.gz | head -n 2  # Nombres de
las columnas
    #Overview of data in ClinVar by gene, dated November 28, 2020
    #Symbol GeneID  Total_submissions       Total_alleles
Submissions_reporting_this_gene
Alleles_reported_Pathogenic_Likely_pathogenic   Gene_MIM_number
Number_uncertain        Number_with_conflicts
```

Como se puede observar en el output de los comandos realizados, este fichero contiene dos headers. Para facilitar la realización del script correspondiente para cargar estos datos en un fichero de datos, se ha decidido eliminar el primer header, que contiene el título del archivo.

```
# Fichero Junio
$ zcat gene_specific_summary_2020-06.txt.gz | tail -n +2 | gzip >
clinvar_gene_2020-06.txt.gz

# Fichero Diciembre
$ zcat gene_specific_summary_2020-12.txt.gz | tail -n +2 | gzip >
clinvar_gene_2020-12.txt.gz
```

Una vez eliminado el título de ambos ficheros, se ha comprobado el número de líneas de datos.

```
# Fichero Junio
$ gunzip -c clinvar_gene_2020-06.txt.gz | wc -l      # Número líneas de
datos
    32954

# Fichero Junio
$ gunzip -c clinvar_gene_2020-12.txt.gz | wc -l      # Número líneas de
datos
    32984
```

Para cargar los datos de estos documentos en las bases de datos SQL será necesario, como se ha hecho en el apartado anterior, crear un programa mediante un script de python. El primer paso es la creación de una tabla para guardar estos datos; *gene_specific*, junto a la descripción de las columnas que la forman. Las primeras cuatro columnas se consideran esenciales, por lo tanto, se han impuesto restricciones fuertes, ignorando los valores nulos. En este caso no se ha considerado esencial para este proyecto usar una primary key en esta tabla.

```
CREATE TABLE IF NOT EXISTS gene_specific (
    gene_symbol VARCHAR(64) NOT NULL,
    gene_id INTEGER NOT NULL,
    total_submissions INTEGER NOT NULL,
    total_alleles INTEGER NOT NULL,
    submissions_gene INTEGER,
    allele_path INTEGER,
    gene_MIM INTEGER,
    number_uncertain INTEGER,
    number_conflicts INTEGER
)
```

Este paso de creación de la tabla var_citations, ha sido incluido en el script del programa, *clinvar_gene.py*, en este caso, volviendo a añadir IF NOT EXIST, para así, poder ejecutarlo exista o no las tabla. Como en el caso anterior, este programa, se ha creado usando el script trabajado en clase, *clinvar_parser.py* como base

y guía pero adaptandolo a las necesidades de la tabla *gene_specific*, por lo tanto se han realizado diferentes modificaciones.

Por ejemplo, el fichero tabular que queremos abrir con el programa esta comprimido, por lo tanto, en este caso si que se ha usado la librería gzip para abrirlo:

```
with gzip.open(clinvar_file,"rt",encoding="utf-8") as cf:
```

Dentro del script de python, se ha determinado las columnas que deseamos introducir, junto a las adecuaciones de su nombre, para finalizar la edición del script.

```
else:
    columnValues = re.split(r"\t",wline)

    for iCol, vCol in enumerate(columnValues):
        if len(vCol) == 0 or vCol == "-":
            columnValues[iCol] = None

    gene_symbol = columnValues[headerMapping["Symbol"]]
    gene_id = int(columnValues[headerMapping["GeneID"]])
    total_submissions =
int(columnValues[headerMapping["Total_submissions"]])
    total_alleles = int(columnValues[headerMapping["Total_alleles"]])
    submissions_gene =
columnValues[headerMapping["Submissions_reporting_this_gene"]]
    allele_path =
columnValues[headerMapping["Alleles_reported_Pathogenic_Likely_pathogenic"]
]
    gene_MIM = columnValues[headerMapping["Gene_MIM_number"]]
    number_uncertain = columnValues[headerMapping["Number_uncertain"]]
    number_conflicts  =
columnValues[headerMapping["Number_with_conflicts"]]

    cur.execute("""
        INSERT INTO gene_specific(
            gene_symbol,
            gene_id,
            total_submissions,
            total_alleles,
            submissions_gene,
            allele_path,
            gene_MIM,
            number_uncertain,
            number_conflicts)
        VALUES(?,?,?,?,?,?,?,?,?)
    """,
(gene_symbol,gene_id,total_submissions,total_alleles,submissions_gene,allel
e_path,gene_MIM,number_uncertain,number_conflicts)
    )
```

Posteriormente, se ha ejecutado el programa creado, *clinvar_gene.py*, añadiendo así la tabla *gene_specific* correspondiente, a los ficheros de datos creados anteriormente, el de Junio y el de Diciembre. A la vez, para comprobar que se han leído y cargado todas las líneas, se ha usado SQLITE3 para contar las líneas de datos de la tabla añadida, que como se observa, es el mismo valor (excluyendo la línea de header), que el observado anteriormente.

```
# DB Junio
python3 clinvar_gene.py clinvar_jun.db clinvar_gene_2020-06.txt.gz

sqlite3 clinvar_jun.db
    -- SQLite
    SELECT COUNT(*)
    FROM gene_specific;
        32953

# DB Diciembre
python3 clinvar_gene.py clinvar_dec.db clinvar_gene_2020-12.txt.gz
sqlite3 clinvar_dec.db
    -- SQLite
    SELECT COUNT(*)
    FROM gene_specific;
        32983
```

Para recapitular, en estos dos apartados iniciales se han diseñado los scripts de python necesarios para cargar los subdirectorios de ClinVar sobre citaciones y sobre estadísticas de genes. Posteriormente, estos programas han cargado las tablas sobre las bases de datos generadas al cargar una release de Clinvar de junio y una de diciembre respectivamente. Ahora ya tenemos las bases de datos preparadas para poder lanzar consultas individuales y combinadas.

PARTE 2

2.1. ¿Cuántas variantes están relacionadas con el gen P53 tomando como referencia el ensamblaje GRCh38?

Para realizar la consulta sobre P53, se tiene que tener en cuenta que en la base de datos aparece como TP53. Esto es porque el gen TP53 proporciona instrucciones para producir la proteína llamada proteína tumoral p53. Esta proteína actúa como supresor de tumores, lo que significa que regula la división celular evitando que las células crezcan y se dividan (proliferen) demasiado rápido o de forma incontrolada. Para saber cuantas variantes están relacionadas con la producción de esta proteína en el ensamblaje GRCh38, se ha usado el script de sql *2_1.sql*.

DB Junio:

Num_Variations
1831

DB Diciembre:

Num_Variations
1994

Al realizar esta consulta, se ha podido ver que en el release de Clinvar de diciembre hay 158 variantes relacionadas con el gen TP53 más que en el release de junio.

2.2. ¿Qué cambio del tipo “single nucleotide variant” es más frecuente, el de una Guanina por una Adenina, o el una Guanina por una Timina? Usad las anotaciones basadas en el ensamblaje GRCh37 para cuantificar y proporcionar los números totales.

En este caso, se ha filtrado las anotaciones basadas en el ensamblaje GRCh37 que tengan una variante de un solo nucleótido, comparando los casos donde hay un cambio de una Guanina por una Adenina y el de una Guanina por una Timina. Teniendo así en cuanta, que Ref es el alelo en el genoma de referencia y Alt es cualquier otro alelo encontrado en ese locus.

DB Junio:

Num_A_to_G
69409

Num_T_to_G
19345

DB Diciembre:

Num_A_to_G
75052

Num_T_to_G
21316

Al hacer la consulta, podemos ver como en ambas DB, es más habitual que una Guanina sea reemplazada por una Adenina que no una Timina.

2.3. ¿Cuáles son los tres genes con un mayor número de inserciones y deleciones? Usa el ensamblaje GRCh37 para cuantificar y proporcionar los números totales.

Para resolver esta cuestión, se ha realizado la consulta de 2_3.sql, con el objetivo de obtener una tabla con cada gene_symbol presente en el DB y el número de veces en los que se ha descrito una inserción o deleción. Se ha usado el recurso LIKE ya que se han detectado diferencias en el uso de mayúsculas inicial entre las dos databases, así, nos aseguramos de incluir todas los valores. Los resultados obtenidos son los siguientes:

DB Junio:

gene_symbol	total_ins_del

gene_symbol	total_ins_del
BRCA2	3902
BRCA1	3438
NF1	1413

DB Diciembre:

gene_symbol	total_ins_del
BRCA2	4071
BRCA1	3591
NF1	1596

Podemos que en ambos releases de ClinVar, los tres genes con más inserciones y deleciones son BRCA2, BRCA1 y NF1. Los dos primeros son genes que producen proteínas que ayudan a reparar el ADN dañado. Todos tenemos dos copias de cada uno de estos genes, una copia heredada de cada padre. BRCA1 y BRCA2 a veces se denominan genes supresores de tumores porque cuando tienen ciertos cambios, llamados variantes (o mutaciones) dañinas (o patógenas), se puede desarrollar cáncer.

Las personas que heredan variantes dañinas en uno de estos genes tienen un mayor riesgo de padecer varios cánceres, sobre todo cáncer de mama y de ovario, pero también varios tipos adicionales de cáncer. Las personas que han heredado una variante dañina en BRCA1 y BRCA2 también tienden a desarrollar cáncer a edades más tempranas que las personas que no tienen dicha variante. Esto se debe a que las células que no tienen proteínas BRCA1 o BRCA2 en funcionamiento pueden crecer sin control y convertirse en cáncer [consultado en https://www.cancer.gov/about-cancer/causes-prevention/genetics/brca-fact-sheet, dic 29 2020].

Por otra parte, el gen NF1 parece tener la función de un regulador negativo de la vía de transducción de señales ras. Las mutaciones en este gen se han relacionado con la neurofibromatosis tipo 1, la leucemia mielomonocítica juvenil y el síndrome de Watson [consultado en https://www.genecards.org/cgi-bin/carddisp.pl?gene=NF1, dic 29 2020].

2.4. ¿Cual es la deleción más común en el cáncer hereditario de mama? Por favor, incluye en la respuesta además en qué genoma de referencia, el número de veces que ocurre, el alelo de referencia y el observado.

Para resolver este enunciado, se ha realizado una consulta donde se ha seleccionado todas las variantes relacionadas con una deleción y un fenotipo de cáncer de mama. Las observaciones resultantes, como se observa a continuación, se han agrupado según el numero de identificación de variante y su ensamblaje.

DB Junio:

dbSNP_id	Total_Occurrencies	assembly	ref_allele	alt_allele
80357575	4	GRCh37	CTTTTT	C
80357575	4	GRCh38	CTTTTT	C

dbSNP_id	Total_Occurrencies	assembly	ref_allele	alt_allele
80359406	4	GRCh37	GAAA	G
80359406	4	GRCh38	GAAA	G
80357522	3	GRCh37	CT	C
80357522	3	GRCh38	CT	C
34570933	2	GRCh37	AT	A
34570933	2	GRCh38	AT	A
56130510	2	GRCh37	TA	T
56130510	2	GRCh38	TA	T

DB Diciembre:

dbSNP_id	Total_Occurrencies	assembly	ref_allele	alt_allele
80357575	4	GRCh37	CTTTTT	C
80357575	4	GRCh38	CTTTTT	C
80359406	4	GRCh37	GAAA	G
80359406	4	GRCh38	GAAA	G
11309117	3	GRCh37	TAAAA	T
11309117	3	GRCh38	TAAAA	T
80357522	3	GRCh37	CT	C
80357522	3	GRCh38	CT	C
397507419	3	GRCh37	CA	C
397507419	3	GRCh38	CA	C

Al observar las 10 deleciones más comunes, vemos como ambos ensamblajes y bases de datos son las clasificadas con los id 80357575 y 80359406 con un total de 4 ocurrencias respectivamente. En un caso el alelo ha pasado de CTTTTT en la referencia a C y en el otro de GAAA a G.

2.5. Ver el identificador de gen y las coordenadas de las variantes del ensamblaje GRCh38 relacionadas con el fenotipo del Acute infantile liver failure due to synthesis defect of mtDNA-encoded proteins.

En este caso se ha seleccionado los genes junto a sus coordenadas que según el ensamblaje GRCh38 están relacionadas con el fenotipo indicado, limitando en la consulta, la observación de solo los 10 primeros valores, para así, poder insertar los valores en esta memoria.

gene_id	gene_symbol	chro_start	chro_stop

gene_id	gene_symbol	chro_start	chro_stop
55687	TRMU	46337925	46337925
55687	TRMU	46353809	46353809
55687	TRMU	46335766	46335766
55687	TRMU	46352316	46352316
55687	TRMU	46335453	46335453
55687	TRMU	46335486	46335486
55687	TRMU	46335618	46335618
55687	TRMU	46335637	46335637
55687	TRMU	46335652	46335652
55687	TRMU	46357216	46357216

DB Junio:

gene_id	gene_symbol	COUNT(*)
55687	TRMU	44

DB Diciembre:

gene_id	gene_symbol	COUNT(*)
55687	TRMU	45

Al analizar el resultado, se puede determinar que solo aparece un gen con variaciones relacionadas con el fenotipo de insuficiencia hepática infantil aguda debido a un defecto de síntesis de proteínas codificadas por mtDNA. Este gen tiene como id 55687 y se conoce como TRMU. Esta es una proteína conservada evolutivamente que participa en la modificación del ARNt mitocondrial y, por lo tanto, es importante para la traducción mitocondrial [consultado en https://omim.org/entry/610230, dic 29 2020]. En la DB de junio presenta 44 variaciones en diferentes coordenadas del cromosoma y en la DB de diciembre, se ha añadido una variante de este gen con diferentes coordenadas.

2.6. Para aquellas variantes de significancia clínica “Pathogenic” o “Likely pathogenic”, recuperar las coordenadas, el alelo de referencia y el alelo alterado para la hemoglobina (HBB) en el assembly GRCh37.

Para poder hacer esta búsqueda en las bases de datos se ha usado la tabla clinical_sig, que describe la significancia clínica de cada variante; con el objetivo de crear una tabla temporal donde estén presentes las columnas que describen la variable junto a la significancia clínica. Posteriormente se ha hecho una consulta seleccionando las entradas con significancia patogénica o probablemente patogénica y que el nombre del gen contenga HBB. A continuación se muestran las 15 primeras líneas de datos obtenidas de cada búsqueda como representación.

DB Junio:

dbSNP_id	chro_start	chro_stop	ref_allele	alt_allele
267607291	5248004	5248005	TGG	T
33958358	5248248	5248248	C	A
63750860	5246884	5246887	CAGC	TGTGG
33918338	5246841	5246841	T	G
33915112	5248172	5248172	T	C
35348864	5248027	5248028	A	AGCC
41417446	5247993	5247995	CAAA	C
33918338	5246841	5246841	T	C
35020585	5246837	5246837	C	G
35693898	5247865	5247865	A	G

DB Diciembre:

dbSNP_id	chro_start	chro_stop	ref_allele	alt_allele
33918338	5246841	5246841	T	C
35020585	5246837	5246837	C	G
41417446	5247993	5247995	CAAA	C
35693898	5247865	5247865	A	G
33930165	5248233	5248233	C	T
33978338	5247985	5247985	A	G
33950507	5248173	5248173	C	T
34160180	5248179	5248181	CCAA	C
34378160	5247994	5247994	A	G
34165323	5247923	5247923	T	C

En la DB de junio se han obtenido un total de 264 diferentes coordenadas de variantes del genoma según el ensamblaje GRCh37 que son patogénicas o probablemente patogénicas al alterar un alelo para la hemoglobina y en la base de datos de diciembre se han obtenido un total de 229, por lo tanto, probablemente por alguna recalificación de la patogeniciedad de las variantes, se ha reducido el número total.

2.7. Calcular el número de variantes del ensamblaje GRCh38 que se encuentren en el cromosoma 13, entre las coordenadas 10,000,000 y 20,000,000.

Para realizar esta consulta se ha seleccionado las variantes del cromosoma 13 con coordenadas que empiezan entre 10000000 y 20000000 y que acaban también entre estos valores, para así incluir todas las

variaciones posibles dentro de estas restricciones. Al contar el número de resultados de la DB de junio, se ha obtenido 16 y en la de diciembre, se ha obtenido el mismo valor. Por lo tanto, se puede ver como en estos meses de diferencia no se ha insertado ninguna entrada en ClinVar con esta descripción.

2.8. Calcular el número de variantes para los cuáles se haya provisto entradas de significancia clínica que no sean inciertas (“Uncertain significance”), del ensamblaje GRCh37, en aquellas variantes relacionadas con BRCA2.

En este caso se ha vuelto a usar la tabla temporal que maneja datos de la tabla variant y la tabla clinical_sig para resolver este problema. Pero esta vez, se ha seleccionado las entradas que no presentan una significancia incierta, 'Uncertain significance' y que en el símbolo de su gen incluye la BRCA2. Obteniendo así, 7357 variantes en la base de datos de junio y 7712 en la de diciembre. Por lo tanto, podemos determinar que entre estos dos meses de describieron 355 variantes con estas restricciones en ClinVar.

2.9. Obtener el listado de pubmed_ids relacionados con las variantes del ensamblaje GRCh38 relacionadas con el fenotipo del glioblastoma.

Para realizar este análisis, se ha usado el recurso de INNER JOIN para trabajar con datos de la tabla variant y de var_citation, unidos en los valores de variation_id. Una vez creada esta tabla, se han seleccionado las entradas que tienen como fuente de la citación a PubMed y que en la descripción fenotípica aparece la palabra Glioblastoma. A continuación se muestran las 15 primeras líneas de datos obtenidas de cada búsqueda como representación, obtenidas al realizar ambas búsquedas.

variation_id	citation_id	gene_symbol
7813	10400993	PTEN
7813	10920277	PTEN
7813	11504908	PTEN
7813	17526800	PTEN
7813	19366826	PTEN
7813	19903786	PTEN
7813	20453058	PTEN
7813	21956414	PTEN
7813	22491738	PTEN
7813	23475934	PTEN
7813	25741868	PTEN
7813	26773036	PTEN
7813	15016963	PTEN
7813	15254419	PTEN
7813	16952599	PTEN

En los resultados obtenidos se puede observar como existen diferentes citation_id por cada variation_id, esto se debe a que diferentes artículos citan la misma variación. En la DB de junio, se han descrito un total de 2123 citaciones en artóculos, de variantes del ensamblaje GRCh38 relacionadas con el fenotipo del glioblastoma. Sin embargo, al analizar la DB de diciembre, algunas de estas citaciones han sido eliminadas, ya que sólo se han detectado un total de 2098.

2.10. Obtén el número de variantes del cromosoma 1 y calcula la frecuencia de mutaciones de este cromosoma, tanto para GRCh37 como para GRCh38. ¿Es esta frecuencia mayor que la del cromosoma 22? ¿Y si lo comparamos con el cromosoma X? Tomad para los cálculos los tamaños cromosómicos disponibles tanto en https://www.ncbi.nlm.nih.gov/grc/human/data?asm=GRCh37.p13 como en https://www.ncbi.nlm.nih.gov/grc/human/data?asm=GRCh38.p13 .

En primer lugar, antes de realizar la consulta en SQL, se han obtenido los cálculos de los tamaños cromosómicos para GRCh37 (publicado el 28-06-2013 en NCBI) y para GRCh38 (publicado el 01-03-2019 en NCBI) en bp.

Human Genome Assembly GRCh37: Cromosona 1: Lenght = 249.250621 bp Cromosoma 22: Lenght = 51.304566 bp Cromosoma X: Lenght = 155.270560 bp

Human Genome Assembly GRCh38: Cromosona 1: Lenght = 248.956422 bp Cromosoma 22: Lenght = 50.818468 bp Cromosoma X: Lenght = 156.040895 bp

Para hacer esta consulta, alteraremos la tabla usada para añadir una columna que represente la longitud del cromosoma, dependiendo de su ensamblaje. Posteriormente, teniendo en cuenta que la frecuencia de mutaciones se puede definir como el total de variaciones en un cromosoma entre la longitud total del cromosoma. Usaremos esta operación para completar esta búsqueda.

DB Junio:

chro	assembly	Total Variations	length	Frequency
1	GRCh37	59399	249.250621	238.310339054281
1	GRCh38	56414	248.956422	226.601907059863
22	GRCh37	15419	51.304566	300.538552455546
22	GRCh38	14218	50.818468	279.780177552775
X	GRCh37	34663	155.27056	223.242577343703
X	GRCh38	31915	156.040895	204.529716392616

DB Diciembre:

chro	assembly	Total Variations	length	Frequency
1	GRCh37	65616	249.250621	263.253105395473
1	GRCh38	62499	248.956422	251.043935713376
22	GRCh37	16605	51.304566	323.6554033027

chro	assembly	Total Variations	length	Frequency
22	GRCh38	15357	50.818468	302.193289258543
X	GRCh37	37783	155.27056	243.336534627041
X	GRCh38	34806	156.040895	223.056910818154

Al analizar el resultado, se observa como en ambos releases de la DB de Clinvar y en ambos ensamblajes, la frecuencia de mutaciones del cromosoma 1 es menor a la frecuencia encontrada en el cromosoma 22 pero a su vez, es mayor que la observada en el cromosoma X.