Universitat Oberta de Catalunya

uoc.edu

Evaluación del rendimiento y comparativa de varios métodos de predicción de patogenicidad y priorización de variantes genéticas.

Víctor Manuel Duarte Rute
Máster Bioinformática y Bioestadística
Área 2. Subárea 6: Estudios de asociación en genómica del cáncer e integración de datos ómicos.

Jaime Sastre Tomàs
Marc Maceira Duch

05/01/2021

B) GNU Free Documentation License (GNU FDL)

C) Copyright

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Evaluación del rendimiento y comparativa de varios métodos de predicción de patogenicidad y priorización de variantes genéticas.*
Nombre del autor:	*Víctor Manuel Duarte Rute*
Nombre del consultor/a:	*Jaime Sastre Tomàs*
Nombre del PRA:	*Marc Maceira Duch*
Fecha de entrega (mm/aaaa):	01/2021
Titulación:	*Máster Bioinformática y Bioestadística*
Área del Trabajo Final:	*Área 2. Subárea 6: Estudios de asociación en genómica del cáncer e integración de datos ómicos.*
Idioma del trabajo:	*Castellano*
Palabras clave	*prioritization, genetic variants, pathogenicity*

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

El análisis terciario dentro de un flujo de trabajo de muestras genómicas procedentes de Secuenciación de Próxima Generación se corresponde con la anotación biológica de las variantes detectadas y su posterior priorización e interpretación con fines clínicos. Para ello existen multitud de herramientas computacionales que intentan predecir mediante diferentes enfoques la potencial patogenicidad de determinadas variantes genéticas, ya sea a través de clasificadores basados en *scores* o de plataformas que prioricen aquéllas con mayor potencial de causar un determinado fenotipo o enfermedad.

En este trabajo se ha llevado a cabo una evaluación y comparación de diferentes métodos de predicción de patogenicidad y priorización de variantes en entornos clínicos, a través de la obtención de métricas estadísticas que evalúen el rendimiento de diferentes clasificadores existentes en la literatura y el análisis pormenorizado de varias plataformas de priorización. Se obtienen varios conjuntos de variantes públicos con los que realizar los diferentes análisis y comparaciones.

Nuestros resultados indican que los predictores con mayor rendimiento para las variantes analizadas son ClinPred, BayesDel, REVEL, VEST4, fathmmMKL y PrimateAI, mientras que por otro lado GenIO se presenta como la mejor plataforma para obtener una lista priorizada de variantes con mayor

patogenicidad. Asimismo, VarCards y OpenCRAVAT destacan por servir de punto de partida para una anotación exhaustiva de las variantes detectadas en el experimento de secuenciación.

Abstract (in English, 250 words or less):

Tertiary analysis within a genomic samples workflow from Next Generation Sequencing corresponds to the biological annotation of the variants detected and their subsequent prioritisation and interpretation for clinical purposes. For this goal, there are many computational tools that try to predict the pathogenicity of certain genetic variants using different approaches, either through score-based classifiers or platforms that prioritise those variants with the greatest potential to cause a certain phenotype or disease.

In this work, an evaluation and comparison of different methods of pathogenicity prediction and variant prioritisation in clinical environments has been carried out, through the computation of statistical metrics that evaluate the performance of different classifiers from the literature and the detailed analysis of several prioritisation platforms. Several sets of public variants are obtained with which to perform the different analyses and comparisons.

Our results show that the predictors with the highest performance for the analysed variants are ClinPred, BayesDel, REVEL, VEST4, fathmmMKL and PrimateAI, while on the other hand GenIO is presented as the best platform to obtain a prioritised list of variants with higher pathogenicity. Likewise, VarCards and OpenCRAVAT stand out as appropriate starting points for the comprehensive annotation of the variants detected in the sequencing experiment.

Índice

Índice de figuras

Índice de tablas

1. Introducción

1.1 Contexto y justificación del Trabajo

Este trabajo se enmarca dentro de una de las etapas clave en los *pipelines* de análisis de datos genómicos procedentes de experimentos de secuenciación masiva, como es la anotación y priorización de variantes genéticas en estudios de interés clínico. Los análisis de datos genómicos de experimentos NGS – *Next Generation Sequencing* – conllevan una serie de pasos y etapas en serie que constituyen lo que se conoce como *pipeline* o *workflow*; dentro de este esquema podemos distinguir varios conjuntos de etapas o bloques de análisis que realizan funciones específicas, como son el análisis primario, que comprende la secuenciación propiamente dicha y todo el diseño del experimento, el análisis secundario, correspondiente a las diferentes etapas de procesamiento bioinformático de los datos (control de calidad, alineamiento respecto al genoma de referencia, detección de variantes), y el análisis terciario, también conocida como la etapa de anotación biológica e interpretación, cuyo papel es aportar información a las variantes detectadas en base a diversos recursos e intentar predecir o interpretar la relevancia clínica de dichas variantes en el contexto de una enfermedad (1).

En esta última etapa de intentar comprender la significancia biológica de dichas variantes se usan métodos y herramientas de anotación, como Ensembl Variant Effect Predictor - VEP - (2), ANNOVAR (3) o SnpEff (4), que llevan a cabo una búsqueda respecto a diferentes recursos de información como bases de datos para aportar información a las variantes detectadas previamente para conocer su función y caracterizarlas, como el cambio de aminoácido provocado, el gen y proteína afectado, el exón donde se localiza o la consecuencia que tiene dicho cambio; este último factor es crucial a la hora de predecir la posible patogenicidad de una variante, y nos podemos encontrar desde polimorfismos en regiones intrónicas o intergénicas, que en principio no resultarían en ningún cambio visible en la proteína (recientemente se está viendo que no, y de hecho las variantes no codificantes pueden jugar un papel importante en la expresión y función de un gen (5)), pasando por cambios que truncan por completo la proteína como las variantes *frameshift* o *in-frame* (modifican la pauta normal de lectura de codones) hasta las mutaciones *missense*, las más sonadas hoy en día y sin duda las que se estudian con mayor énfasis por ser la potencial causa de un gran número de enfermedades mendelianas (6).

En base a toda esta información la labor del clínico es interpretar estos datos y, en función de la sintomatología y las características del paciente, encontrar la variante o conjunto de variantes causante de la enfermedad que se está analizando. Por tanto, es una etapa imprescindible en el análisis de datos genómicos con fines clínicos, con una gran diversidad de herramientas que pretenden solucionar este gran reto con muchísimas cuestiones a resolver (7).

Esta última fase del proceso es generalmente la que se está encontrando con diversos obstáculos con los continuos avances en computación, *big data* y tecnología de la secuenciación, que están permitiendo que multitud de

laboratorios clínicos en el mundo empiecen a plantearse la secuenciación masiva como solución al diagnóstico, prevención y tratamiento de multitud de enfermedades; esto hace que cada vez sea más difícil con las herramientas existentes interpretar adecuadamente la ingente cantidad de variantes genéticas que se pueden obtener en un experimento normal de secuenciación de exoma completo, convirtiéndose en el clásico problema de encontrar una aguja en un pajar. Por ello en los últimos años se vienen desarrollando multitud de algoritmos y métodos para predecir la potencial patogenicidad de variantes genéticas en base a diferentes factores, como la frecuencia alélica en la población, el grado de conservación de esa zona genómica respecto a otras secuencias homólogas o la posible variación en la estabilidad de la proteína resultante, por poner varios ejemplos (8).

La elección de este proyecto enmarcado en la interpretación clínica de variantes genéticas se basa en varias razones. En primer lugar, tal y como se ha comentado, la continua mejora de las tecnologías de secuenciación genómica está permitiendo la obtención de una cantidad cada vez mayor de datos genéticos en espera de poder ser analizados e interpretados, siendo actualmente el principal cuello de botella de todo el proceso de análisis. El hecho de que las tecnologías de secuenciación de genoma completo estén reduciendo sus costes por debajo de los 1000$ hace que buena parte de los laboratorios clínicos se estén enfocando más en esta práctica para la caracterización genética de multitud de enfermedades y pacientes, especialmente con aquellos donde la secuenciación clásica o por paneles de genes no reporta grandes resultados (9). La secuenciación de un genoma completo implica la obtención de al menos varios conjuntos de datos que pueden alcanzar los 100 GB de almacenamiento, a los que podemos sumar varios ficheros intermedios correspondientes al alineamiento y demás tipos de procesamientos que engrosan esa cantidad; sin embargo, el tamaño de los ficheros resultantes no es lo que verdaderamente asusta, sino la enorme cantidad de variantes detectadas en una única muestra, del orden del millón de cambios respecto al genoma de referencia. Esto convierte la tarea de encontrar la variante responsable de una determinada enfermedad en un clásico problema de buscar una aguja en un pajar, ya que el 99,9 % de los cambios detectados no tienen ninguna consecuencia visible en el organismo, ya sea porque los cambios ocurren en heterocigosis, o se corresponden con variantes sinónimas donde el cambio de nucleótido no genera ninguna variación en la proteína, etc. Por todo ello surge la necesidad de enfrentarse a este reto de interpretar correctamente las variantes genéticas y caracterizar la relevancia clínica que poseen, lo que nos ha llevado a este proyecto en el que se pretende evaluar diferentes plataformas encargadas de esta tarea de priorización clínica de variantes, aportando un valor añadido al facilitar posteriormente la tarea de médicos y/o investigadores cuando necesiten abordar este análisis con una serie de herramientas que consideremos más adecuadas (1).

En segundo lugar, la falta de una adecuada comparación y evaluación de este tipo de herramientas ha hecho que esta área en concreto sea muy interesante para plantear el proyecto, ya que podemos completar el hueco existente en el campo mediante la realización de un *benchmarking* y resumir las ventajas y los inconvenientes de diferentes plataformas de priorización de variantes. Es cierto

que ya existen revisiones que evalúan el rendimiento de diferentes métodos de clasificación de variantes patógenas, cuyo desarrollo viene haciéndose desde mediados de la década de los 2000, a medida que fueron apareciendo los primeros predictores (10–15); sin embargo, a nuestro juicio son trabajos que pueden haber quedado estancados por el paso del tiempo, ya que cada año surgen nuevos algoritmos en este tema que quedan fuera del radar de estos análisis, o que resultan poco exhaustivos en términos de métricas estadísticas calculadas o conjuntos de datos testeados. En este trabajo por tanto se pretende mejorar estas revisiones ofreciendo una visión nueva con diferentes conjuntos de datos más actualizados y con una mayor relevancia en cuanto a las métricas estadísticas calculadas (16). Del mismo modo, en años recientes se han desarrollado herramientas web muy prometedoras que intentan convertirse en la plataforma de apoyo de investigadores y personal clínico para la interpretación genética de una enfermedad, ofreciendo funcionalidades que incluyen desde la simple anotación de las variantes hasta la inclusión de sintomatología o términos fenotípicos para priorizar el conjunto de variantes obtenidas en el experimento (17). Este tipo de herramientas aún no disponen de una evaluación que resuma sus fortalezas y debilidades para conocer cuáles son aquellas que funcionan y cumplen con su objetivo de forma más apropiada, por lo que este trabajo puede servir de guía o punto de partida para enfrentarse a este tipo de retos en la última etapa del flujo de análisis de datos NGS.

En este trabajo por tanto se pretende evaluar y comparar diversos métodos de predicción de patogenicidad y priorización en variantes genéticas para ver cómo se comportan ante determinados conjuntos de datos, y concluir qué método o conjunto de métodos es el adecuado para llevar a cabo esta tarea. La comparativa se desarrollará a través de diferentes análisis bioinformáticos y gráficos que apoyen las conclusiones realizadas, en los que se hará uso de diversas herramientas como son: lenguaje de programación *Bash* (18) y aplicaciones de línea de comandos como *gawk* (19), lenguaje de análisis estadístico R para la realización de las comparativas en las predicciones con paquetes como *ROCit* (20) o *caret* (21). La relevancia de este reto en el contexto clínico va a seguir aumentando en los próximos años a medida que la medicina de precisión se implemente definitivamente en los hospitales y laboratorios de diagnóstico clínico en todo el mundo.

1.2 Objetivos del Trabajo

- Objetivos generales

    - Elegir diferentes métodos de predicción de patogenicidad en base a sus características internas, recabar conjuntos de datos con información clínica fiable y obtener las clasificaciones para cada variante (patogenicidad o neutralidad).
    - Calcular diferentes métricas estadísticas para evaluar las predicciones (ROC, precisión, sensibilidad, etc.).
    - Elegir varias herramientas de priorización de variantes y realizar un análisis con los conjuntos de datos escogidos, midiendo y revisando

diferentes aspectos de su uso (concordancia entre métodos, facilidad de implementación, etc.).

- Objetivos específicos

    - Revisar la literatura existente sobre los diferentes métodos de predicción de patogenicidad basados en score desarrollados hasta la fecha, además de las diferentes revisiones y/o *benchmarkings* realizados para compararlos entre sí.
    - Elegir varios de estos métodos, al menos cinco, basados en diferentes factores o enfoques algorítmicos, basándonos en el conocimiento extraído del primer punto.
    - Buscar conjuntos de datos apropiados para evaluar estas herramientas, con información contrastada y fiable.
    - Obtener las diferentes predicciones, tanto numéricas como categóricas, para el conjunto o conjunto de variantes escogido como *dataset* de prueba de todos los predictores a analizar.
    - Calcular, según el *dataset*, diferentes métricas estadísticas que midan el rendimiento de los predictores, ya sea mediante comparación con información clínica (obtención de matriz de decisión, curva ROC, número de falsos positivos, etc.) o con cálculo de concordancias entre métodos.
    - Revisar la literatura sobre los métodos más recientes de priorización de variantes basados en fenotipos, ya sean basados en servidor web o en línea de comandos.
    - Escoger varias plataformas de priorización de variantes, al menos dos, basándonos en factores como la novedad, el nivel de uso de la comunidad clínica o su facilidad de implementación en nuestros conjuntos de datos.
    - Analizar dichas herramientas con nuestros conjuntos de datos del anterior bloque, obteniendo los resultados en forma de tablas o gráficas.
    - Analizar el rendimiento de las plataformas escogidas mediante el cálculo de ciertos estadísticos o la comparación de la lista priorizada de variantes, además de otros aspectos subjetivos como la facilidad de uso, diseño, enfoque, exhaustividad de los resultados ofrecidos, etc.

1.3 Enfoque y método seguido

Cuando se propuso el tema de la evaluación de métodos de priorización de variantes no habíamos decidido aún la metodología y el enfoque que íbamos a seguir, ya que todavía no se había discutido en profundidad sobre ello. Inicialmente se llevó a cabo un estudio preliminar de la literatura acerca del tema para conocer un poco mejor el estado del arte y decidir los pasos a seguir, que dependían en mayor o menor medida de la variedad de métodos existentes, su disponibilidad para implementarlo en línea de comandos o mediante servidor web o su nivel de implantación en los flujos de análisis de datos NGS con fines clínicos.

Un primer enfoque se basaría en escoger herramientas de priorización según el tipo de análisis interno que realiza para predecir aquellas variantes que puedan ser potencialmente patogénicas, es decir, clasificar los métodos en función del análisis y compararlos para ver cuál es con el que se obtienen los mejores resultados. En cuanto a los diferentes métodos de predicción podríamos encontrarnos con el cálculo de la similitud de la secuencia modificada mediante homología, el análisis fisicoquímico del aminoácido modificado para comprobar la estabilidad futura de la proteína, la comprobación de la frecuencia alélica en una población determinada, etc.; esta estrategia, aunque a priori parezca tener su lógica, no creemos que sea la idónea debido a que, tras revisar la bibliografía correspondiente y analizar los diferentes tipos de métodos de priorización que existen, la diversidad de herramientas existentes en el mercado no puede ser catalogada en los puntos que hemos mencionado previamente. Es cierto que muchos de ellos utilizan o se basan en uno o un conjunto de estos métodos, pero no se caracterizan por dicho método exclusivamente. Sería muy complicado separar los métodos de priorización basándonos en uno de estos factores, más teniendo en cuenta que con el paso del tiempo estas plataformas están integrando cada vez más recursos de información para convertirse en herramientas de apoyo muy exhaustivas.

En segundo lugar, otro planteamiento de enfoque para este proyecto sería el de llevar a cabo un análisis pormenorizado de todos los métodos de predicción de patogenicidad basados en *scores*, comparar su rendimiento y evaluar cuáles son los más eficaces en la tarea de priorizar variantes genéticas; no obstante, esta estrategia no sería exclusiva de este trabajo, ya que como hemos comprobado en la literatura ya se han realizado recientemente diferentes *benchmarkings* que evalúan muchos de estos métodos (13,14), lo que hace que un enfoque dedicado exclusivamente a esto no contribuya demasiado al conocimiento existente. Por esta razón, no consideramos tampoco esta estrategia como la más adecuada para enfocar nuestro proyecto.

Como tercera y última opción, con el objetivo de realizar un trabajo que sea al mismo tiempo exhaustivo y añada valor añadido al conjunto de análisis existentes en la bibliografía, planteamos un enfoque basado en dos vertientes: por un lado, llevar a cabo un análisis de ciertos predictores de patogenicidad basados en scores, ya comentados previamente, pero con la particularidad de escoger herramientas más novedosas y con algoritmos diferentes entre sí, además de utilizar conjuntos de datos optimizados con nueva información sobre las variantes publicadas (ClinVar (22), gnomAD (23), Genome in a Bottle (24), etc.), para así mejorar la perspectiva de los *benchmarkings* desarrollados en los últimos años; por otro lado, para complementar este análisis decidimos que sería buena idea realizar una evaluación de algunas de las recientes plataformas de las que hemos hablado anteriormente, cuyo valor recae en que carecen de una comparación adecuada para que puedan asentarse definitivamente en el mercado de la investigación clínica. Estas plataformas, cuya característica principal es la inclusión de propiedades fenotípicas de los pacientes para la priorización de variantes, no han sido evaluadas mediante una evaluación sólida y fiable, debido especialmente a su reciente desarrollo y publicación en el campo, por lo que añadir este enfoque en nuestro proyecto le

añadiría un gran valor y podría sentar las bases para la inclusión y asentamiento definitivo de estas herramientas en el marco clínico. Por todo ello, esta será nuestra estrategia definitiva para enfocar el reto planteado al inicio, el de evaluar y comparar diferentes métodos de priorización de variantes procedentes de secuenciación masiva con fines clínicos, cuyo planteamiento se ha desarrollado previamente en el apartado de los objetivos.

1.4 Planificación del Trabajo

Se desglosan a continuación todas las tareas correspondientes a cada uno de los objetivos específicos planteados anteriormente, con una breve descripción lo suficientemente completa como para que se comprenda y podamos ajustar la duración y los hitos adecuados.

1.4.1 Tareas

i. Revisar la literatura existente sobre los diferentes métodos de predicción de patogenicidad basados en score desarrollados hasta la fecha, además de las diferentes revisiones y/o benchmarkings realizados para compararlos entre sí.

- Reunir bibliografía referente a los métodos de predicción de patogenicidad existentes, además de revisiones sobre el tema y *benchmarkings* publicados hasta la fecha. Duración: 3 días.
- Revisar en profundidad la literatura reunida en el anterior punto. Duración: 5 días.

ii. Elegir varios de estos métodos, al menos cinco, basados en diferentes factores o enfoques algorítmicos, basándonos en el conocimiento extraído del primer punto.

- Escoger varios métodos de todos los analizados en la tarea anterior de revisión, especialmente catalogándolos en grupos según su algoritmo interno de predicción. Duración: 1 día.
- Describir y caracterizar los grupos de métodos mencionados en el anterior punto. Duración: 5 días.

iii. Buscar conjuntos de datos apropiados para evaluar estas herramientas, con información contrastada y fiable.

- Revisar la bibliografía relacionada con evaluaciones de este tipo de métodos para ver qué conjuntos de datos se suelen utilizar. Duración: 2 días.
- Escoger uno o varios conjuntos de datos apropiados para la evaluación de los predictores de patogenicidad. Duración: 1 día.

- Preprocesar o filtrar las variantes de estos conjuntos de datos para adecuar los ficheros a los métodos escogidos según sea necesario (anotar variantes, filtrar por tipo, etc.). Duración: 2 días.

iv. *Obtener las diferentes predicciones, tanto numéricas como categóricas, para el conjunto o conjunto de variantes escogido como dataset de prueba de todos los predictores a analizar.*

- Llevar a cabo los análisis correspondientes y obtener las predicciones para los métodos escogidos respecto a nuestros conjuntos de datos. Duración: 2 días.
- Preprocesar los resultados obtenidos anteriormente para mejorar o modificar el formato de las predicciones, según sean numéricas o categóricas. Duración: 2 días.

v. *Calcular, según el dataset, diferentes métricas estadísticas que midan el rendimiento de los predictores, ya sea mediante comparación con información clínica (obtención de matriz de decisión, curva ROC, número de falsos positivos, etc.) o con cálculo de con concordancias entre métodos.*

- Escoger y describir varias métricas estadísticas adecuadas en la literatura sobre el tema para evaluar nuestras predicciones. Duración: 1 día.
- Analizar y comparar el rendimiento de los predictores en base a estas métricas escogidas. Duración: 2 días.
- Crear si fuera posible gráficos que ayuden a comprender de manera visual la evaluación que estamos realizando. Duración: 1 día.

vi. *Revisar la literatura sobre los métodos más recientes de priorización de variantes basados en fenotipos, ya sean basados en servidor web o en línea de comandos.*

- Reunir y revisar la bibliografía referente a los métodos de priorización de variantes basados en fenotipos desarrollados hasta la fecha. Duración: 3 días.

vii. *Escoger varias plataformas de priorización de variantes, al menos dos, basándonos en factores como la novedad, el nivel de uso de la comunidad clínica o su facilidad de implementación en nuestros conjuntos de datos.*

- Escoger y describir varias de estas plataformas, explicando sus características y funcionalidades. Duración: 3 días.

viii. *Analizar dichas herramientas con nuestros conjuntos de datos del anterior bloque, obteniendo los resultados en forma de tablas o gráficas.*

- Llevar a cabo los análisis correspondientes en cada una de las plataformas elegidas, siguiendo los pasos que se indiquen. Duración: 3 días.
- Obtener resultados de los análisis realizados, ya sean a modo de gráficas, tablas o capturas de pantalla donde se observe la plataforma en caso de que sea web. Duración: 2 días.

ix. Analizar el rendimiento de las plataformas escogidas mediante el cálculo de ciertos estadísticos o la comparación de la lista priorizada de variantes, además de otros aspectos subjetivos como la facilidad de uso, diseño, enfoque, exhaustividad de los resultados ofrecidos, etc.

- Medir mediante determinadas métricas estadísticas el rendimiento de estas plataformas de priorización y comparar sus resultados según corresponda. Duración: 2 días.
- Analizar y caracterizar el uso de estas herramientas en base a determinados factores para evaluar y decidir cuáles resultan más precisas o beneficiosas para la práctica clínica. Duración: 3 días.

x. Elaboración de entregables finales

- Redacción y entrega de memoria del proyecto. Duración: 50 días.
- Obtención de la presentación virtual para la defensa pública. Duración: 5 días.
- Redacción de la autoevaluación. Duración: 2 días.

1.4.2 Calendario

Adjuntamos una captura del diagrama Gantt realizado mediante la herramienta GanttProject (25) en la Figura 1, en la que se pueden observar las diferentes tareas definidas anteriormente organizadas por duración y por bloque (cada color representa un objetivo específico donde puede haber varias tareas). Anteriormente se desarrolló un organigrama previo donde las tareas del segundo bloque (referentes a las plataformas de priorización, tareas 7, 8 y 9, además de la finalización de los entregables finales de la tarea 10) se realizaban justo después de las del primer bloque, pero fue necesario hacer una reorganización para la entrega de la PEC 1. Por razones de tiempo y la gran carga de trabajo que conllevaba el primer bloque de análisis del proyecto, y que claramente no se supo prever, no se completó la realización de los diferentes objetivos específicos del bloque de análisis de priorización de variantes, por lo que la finalización y entrega de los hitos entregables queda de la siguiente manera en función de las PEC de seguimiento:

- PEC 2

    - Reunir bibliografía referente a los métodos de predicción de patogenicidad existentes, además de revisiones sobre el tema y *benchmarkings* publicados hasta la fecha.
    - Revisar en profundidad la literatura reunida en el anterior punto.
    - Escoger varios métodos de todos los analizados en la tarea anterior de revisión, especialmente catalogándolos en grupos según su algoritmo interno de predicción.
    - Describir y caracterizar los grupos de métodos mencionados en el anterior punto.
    - Revisar la bibliografía relacionada con evaluaciones de este tipo de métodos para ver qué conjuntos de datos se suelen utilizar.
    - Escoger uno o varios conjuntos de datos apropiados para la evaluación de los predictores de patogenicidad.
    - Preprocesar o filtrar las variantes de estos conjuntos de datos para adecuar los ficheros a los métodos escogidos según sea necesario (anotar variantes, filtrar por tipo, etc.).
    - Llevar a cabo los análisis correspondientes y obtener las predicciones correspondientes a los métodos escogidos respecto a nuestros conjuntos de datos.
    - Preprocesar los resultados obtenidos anteriormente para mejorar o modificar el formato de las predicciones, según sean numéricas o categóricas.
    - Escoger y describir varias métricas estadísticas adecuadas en la literatura sobre el tema para evaluar nuestras predicciones.
    - Analizar y comparar el rendimiento de los predictores en base a estas métricas escogidas.
    - Crear si fuera posible gráficos que ayuden a comprender de manera visual la evaluación que estamos realizando.

- PEC 3

    - Reunir y revisar la bibliografía referente a los métodos de priorización de variantes basados en fenotipos desarrollados hasta la fecha.
    - Escoger y describir varias de estas plataformas, explicando sus características y funcionalidades.
    - Llevar a cabo los análisis correspondientes en cada una de las plataformas elegidas, siguiendo los pasos que se indiquen.
    - Obtener resultados de los análisis realizados, ya sean a modo de gráficas, tablas o capturas de pantalla donde se observe la plataforma en caso de que sea web.

- Medir mediante determinadas métricas estadísticas el rendimiento de estas plataformas de priorización y comparar sus resultados según corresponda.
- Analizar y caracterizar el uso de estas herramientas en base a determinados factores para evaluar y decidir cuáles resultan más precisas o beneficiosas para la práctica clínica.

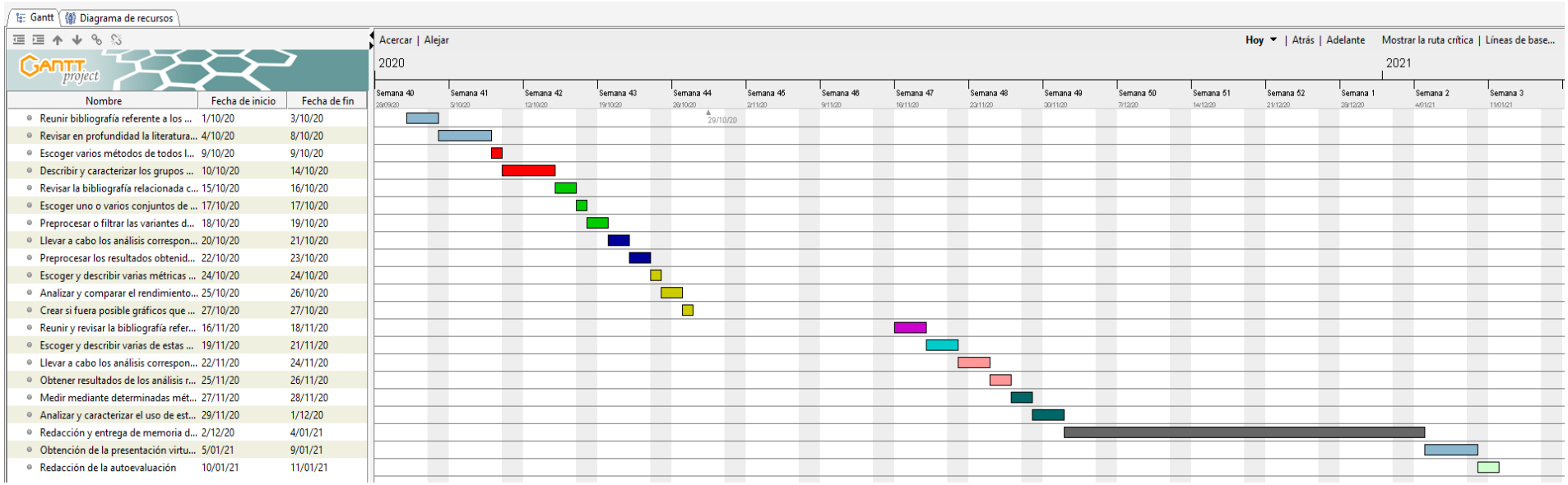


Figura 1. Diagrama Gantt con la planificación temporal de las distintas tareas que componen este trabajo.

1.4.3 Hitos

En este apartado procedemos a enumerar los diferentes hitos que debemos obtener en varias de las tareas descritas, una serie de resultados tangibles con los que se pueda medir el progreso del proyecto. Simplemente mencionaremos aquellas tareas donde al finalizarlas deberíamos obtener un pequeño documento con los resultados que se describen.

i. Describir y caracterizar los grupos de métodos mencionados en el anterior punto.
ii. Escoger uno o varios conjuntos de datos apropiados para la evaluación de los predictores de patogenicidad.
iii. Llevar a cabo los análisis correspondientes y obtener las predicciones correspondientes a los métodos escogidos respecto a nuestros conjuntos de datos.
iv. Escoger y describir varias métricas estadísticas adecuadas en la literatura sobre el tema para evaluar nuestras predicciones.
v. Crear si fuera posible gráficos que ayuden a comprender de manera visual la evaluación que estamos realizando.
vi. Escoger y describir varias de estas plataformas, explicando sus características y funcionalidades.
vii. Obtener resultados de los análisis realizados, ya sean a modo de gráficas, tablas o capturas de pantalla donde se observe la plataforma en caso de que sea web.

1.5 Breve sumario de productos obtenidos

A continuación se resumirán los diferentes productos finales que se obtienen con este trabajo, aunque el detalle exacto de cada uno de ellos se irá explicando a lo largo de los siguientes capítulos de la memoria.

- Memoria

El presente documento servirá como memoria del Trabajo Fin de Máster, realizado a medida de los requerimientos del Plan Docente de la asignatura, donde iremos recogiendo a través de diferentes secciones el planteamiento, desarrollo y obtención de los diferentes resultados del trabajo, para finalmente acabar con una serie de conclusiones que resuman perfectamente los análisis realizados y los resultados obtenidos.

- Evaluaciones predictores de patogenicidad

Se desarrollan como parte del trabajo una serie de informes en formato R Markdown (ficheros en formato Rmd y HTML) con los diferentes scripts encargados de analizar las predicciones con los métodos elegidos para los conjuntos de datos, obteniendo las métricas estadísticas y los gráficos correspondientes.

- Predicciones y anotaciones

Se entregan también en forma de ficheros anexos las anotaciones obtenidas mediante los diferentes métodos de predicción y plataformas de priorización de variantes, en su mayoría en forma de tablas y ficheros en formato Excel o tabulado.

- Presentación virtual

Reflejaremos los resultados obtenidos y las explicaciones necesarias en una presentación PowerPoint para su defensa pública al finalizar la asignatura

- Autoevaluación del proyecto

Se corresponde con una evaluación final de nuestro propio trabajo al final de la presente memoria, calificando las conclusiones obtenidas y proponiendo posibles mejoras a futuro.

1.6 Breve descripción de los otros capítulos de la memoria

- Materiales y métodos

En este primer apartado se describirán todos los materiales y métodos usados en el trabajo, desde las características técnicas del equipo utilizado para el análisis hasta los diferentes algoritmos, paquetes de R, métodos de predicción, métricas estadísticas, conjuntos de datos y plataformas de priorización evaluados.

- Resultados y discusión:

Descripción de los resultados obtenidos en los diferentes análisis de evaluación de herramientas, mostrando gráficas, tablas y capturas de pantalla para explicar los datos; asimismo, se irán describiendo las diferentes discusiones extraídas en base a dichos resultados.

- Conclusiones

Se exponen las conclusiones finales del trabajo, resumiendo las discusiones anteriormente expuestas y analizando mediante una autoevaluación los puntos a mejorar de nuestro trabajo y las líneas de investigación futuras.

2. Materiales y métodos

2.1 Sumario de recursos técnicos

Al tratarse de un trabajo con un componente importante de revisión de distintos métodos de análisis de datos, en este apartado procederemos a hacer un repaso de las diferentes herramientas y/o plataformas a evaluar tratando de describir sus principales características, como el algoritmo que siguen para llevar a cabo la predicción, el grupo de técnicas al que pertenecen dentro del campo, su aplicación para diferentes tipos de variantes o contextos o el tipo de propiedad funcional que intentan predecir. Asimismo, se describirán los conjuntos de datos que utilizaremos como base para realizar los diferentes análisis de predicción e interpretación, revisando sus ventajas o por qué suelen usarse en este tipo de trabajos de *benchmarking*. Finalmente, los cálculos y métricas estadísticas que usaremos para evaluar las predicciones en el primer bloque del trabajo serán también analizadas, teniendo en cuenta sus potenciales beneficios como puntos de partida para comparar diferentes técnicas clasificatorias, como es el caso de los predictores de patogenicidad que se analizarán en este trabajo, cuyo resultado final suele ser una etiqueta que clasifica las variantes introducidas en patógenas/deletéreas o neutras/benignas.

En primer lugar procedemos a enumerar y describir los diferentes métodos y recursos técnicos utilizados en este trabajo. Tras escoger los métodos de predicción y los conjuntos de datos a testar, que serán descritos a lo largo de los siguientes apartados, se hizo uso la terminal de comandos de Linux para realizar diferentes preprocesamientos de los ficheros de variantes y la posterior obtención de las clasificaciones de los predictores; concretamente, se usaron diferentes comandos de *gawk* (19) y *grep* (18) para filtrar las variantes a analizar y la herramienta de búsqueda de anotaciones del repositorio dbNSFP (26) en lenguaje Java (27), mediante la cual se obtuvieron las diferentes predicciones para cada variante seleccionando las columnas apropiadas. Se descargó la herramienta de dbNSFP junto con su base de datos completa, de aproximadamente 25 GB de tamaño, por lo que se necesitó un buen espacio de almacenamiento en el disco duro. El equipo en cuestión usado para todos los análisis cuenta con las siguientes prestaciones: procesador Intel Core i5-9300H 2.40 GHz, con 4 núcleos y 8 hilos, memoria RAM de 8 GB y disco duro SSD de 500 GB de almacenamiento. Todos los comandos de *bash* fueron realizados en la terminal de Ubuntu instalada con el programa WSL2 dentro del propio sistema operativo de Windows 10 (28).

Para los análisis comparativos de las predicciones se utilizaron diversos informes en R Markdown (29) con el lenguaje de análisis estadístico R (30), en su versión 4.0.0 de Windows, y su entorno de desarrollo RStudio (31), en su versión Desktop 1.2.5042. Se hacen uso de los siguientes paquetes de R para los diferentes análisis estadísticos y gráficos a realizar: *caret* (21), en su versión 6.0-86; *ROCit* (20), en su versión 2.1.1; *mltools* (32), en su versión 0.3.5; *knitr* (33), en su versión 1.28; *kableExtra* (34), en su versión 1.3.1; *ggplot* (35), en su versión 3.3.0; y *reshape* (36), en su versión 0.8.8.

Finalmente, aunque las características de las plataformas de priorización e interpretación se detallarán en posteriores apartados, comentamos brevemente que los diferentes análisis en dichas herramientas se realizaron mediante sus servidores en la web, sin la necesidad de instalar ningún software en local, por tanto, en este segundo bloque del trabajo se hace uso sólo y exclusivamente de un navegador como Google Chrome para acceder a las plataformas y obtener los resultados correspondientes.

2.2 Métodos de predicción de patogenicidad

La primera tarea consistió en revisar la bibliografía existente hasta la fecha sobre diferentes algoritmos *in silico* de predicción de patogenicidad de variantes, tema principal del *benchmarking* que se realizará en este trabajo. Se pretendió buscar la mayor exhaustividad posible al elegir los métodos a analizar, teniendo en cuenta factores como el tipo de algoritmo interno en el que estén basados (enfoque bayesiano, alineamiento de secuencias, modelos de *machine learning* tipo SVM, *deep learning*, etc), el tipo de variables o características con las que intentan predecir la patogenicidad (secuencia, función biológica, conservación, entre otros) o su enfoque dependiendo de si incluyen información procedente de otros predictores, lo que conocemos como *ensemble scores*, o si intentan realizar la predicción a partir de su propio algoritmo.

A continuación, procederemos a describir los diferentes métodos escogidos anotando sus principales características, como el algoritmo interno que utilizan o los conjuntos de variantes que usan para entrenar y medir el rendimiento de su clasificador, algo que como veremos más adelante será un factor a tener en cuenta para medir con objetividad los resultados obtenidos, y por lo que intentaremos utilizar diferentes conjuntos de datos para que las métricas sean lo más fiable posible. En la Tabla 1 se muestra una tabla con los predictores escogidos ordenados según el tipo de método al que pertenezcan, detallando además para consultar de forma rápida el algoritmo interno que utilizan para construir el modelo de predicción.

Tabla 1. Lista de predictores de patogenicidad analizados, detallando el algoritmo de predicción que utilizan y el tipo de método donde se agrupan.

Predictor	Algoritmo	Tipo
SIFT	Position-specific scoring matrix	Funcional
VEST	Random Forest	Funcional
DEOGEN2	Random Forest	Funcional
LIST-S2	Local Sequence Identity	Funcional
PrimateAI	Deep Neural Network	Funcional
CADD	Linear kernel Support Vector Machine (SVM)	Ensemble
DANN	Deep Neural Network	Ensemble
FATHMM-MKL	Multiple kernel learning	Ensemble
MetaLR	Logistic Regression (LR)	Ensemble

MetaSVM	Support Vector Machine (SVM)	Ensemble
REVEL	Random Forest	Ensemble
Eigen	Hierarchical model	Ensemble
BayesDel	Naïve weighted Bayesian approach	Ensemble
ClinPred	Random forest/Gradient Boosting	Ensemble
GERP++	Maximum likelihood evolutionary rate estimation	Conservación
PhyloP	Phylogenetic Hidden Markov Model	Conservación

2.2.1 Predictores funcionales sin enfoque ensemble

- SIFT

El algoritmo *Sorting Intolerant From Tolerant* (SIFT), uno de los primeros métodos de predicción de patogenicidad desarrollados, ha sido una herramienta ampliamente utilizada durante años como predictor por defecto para evaluar la patogenicidad de variantes de cambio de aminoácido, las conocidas como *missense*. Este método se basa simplemente en construir una matriz de posiciones en la que se representan todos los posibles cambios de aminoácido en las distintas posiciones de la proteína, mostrándolo a través de un alineamiento de secuencias similares. Posteriormente el algoritmo calcula la probabilidad de que un cambio determinado sea patógeno o no dependiendo de si esa posición se encuentra altamente conservada en el alineamiento, dando a entender que se trataría de un aminoácido esencial y su cambio provocaría la pérdida de función de la proteína; además esto también tiene en cuenta las propiedades fisicoquímicas de las moléculas implicadas, ya que se considera que unas características similares pueden hacer que el cambio sea tolerado por la proteína (37).

- VEST

El *Variant Effect Scoring Tool* (VEST), desarrollado en 2013, es un método de predicción que lleva a cabo la clasificación de patogenicidad basándose en numerosas variables o *features* obtenidas de la herramienta SNVBox (38), hasta 86 campos relativos a las características fisicoquímicas del cambio de aminoácido, la presencia de dominios o zonas importantes en la función de la proteína y otras diversas anotaciones basadas en su posición genómica. A partir de estas variables utilizó para su entrenamiento un conjunto de variantes negativas procedentes del repositorio HGMD (caracterizadas como patógenas) (39) y otro benigno del proyecto de secuenciación de exomas (ESP, *Exome Sequencing Project*) (40), con frecuencias alélicas superiores al 1 %, mientras que se implementó como algoritmo interno un clasificador *Random Forest* (41).

- DEOGEN2

DEOGEN2, implementado en 2017 como evolución del predictor anterior DEOGEN, es un algoritmo basado en *Random Forest* que clasifica mediante puntuación numérica la patogenicidad de variantes no sinónimas (*missense*), utilizando para ello múltiples variables basadas en el contexto genómico de la variante, las propiedades fisicoquímicas de los aminoácidos implicados, etc. El

avance que presenta esta herramienta puede deberse a que incluye información sobre las características evolutivas de la proteína o los métodos de interacción con otras biomoléculas, por lo que el conjunto de *features* que utiliza es bastante completo. Utiliza para el entrenamiento del modelo variantes procedentes del repositorio *Humsavar*, perteneciente a la base de datos de secuencias proteicas por excelencia, UniProt (42).

- LIST-S2

Predictor desarrollado en 2020 como actualización de su predecesor LIST, cuya característica más importante es el cálculo de diversas métricas relacionadas con el contexto evolutivo de la secuencia, como son la *local sequence identity* y el *shared taxa*, que permiten medir cómo de grave es el cambio de aminoácido en una posición determinada según la conservación de dicho aminoácido en especies evolutivamente cercanas; cuantas más especies compartan esa posición más patogenicidad puede provocar la variante, medidas a través de alineamientos múltiples. Para construir la herramienta se usan *datasets* procedentes de los experimentos de secuenciación masiva ExAC y gnomAD, además de variantes catalogadas como patógenas por la base de datos UniProt (43).

- PrimateAI

Herramienta muy novedosa desarrollada por primera vez en 2018 capaz de clasificar variantes según su potencial patogenicidad basándose en el propio efecto de dichos cambios en varias especies de primates altamente relacionados evolutivamente con el humano (entre ellas el chimpancé o el orangután). Utiliza conjuntos de datos de ClinVar y especialmente de repositorios masivos para obtener un gran *dataset* de variantes consideradas benignas según su frecuencia alélica; esto hace que su capacidad de detectar o predecir la tolerancia de variantes genéticas se ve ampliamente mejorada respecto a la mayoría de predictores del mercado. Como algoritmo interno utiliza un conjunto de redes neuronales convolucionales profundas, englobadas dentro del campo del *deep learning*, con las que es capaz de hacer la predicción utilizando como *features* la propia secuencia genómica primaria (cada posición representada en una variable diferente) y algunas propiedades estructurales de la proteína implicada (44).

2.2.2 Predictores funcionales basados en método ensemble

- CADD

El primer algoritmo que vamos a ver dentro del grupo con enfoque ensemble es CADD, o *Combined Annotation-Dependent Depletion*, desarrollado por primera vez en 2013, cuya peculiaridad más importante es el uso como variables de otros algoritmos de predicción existentes, como SIFT o PolyPhen (45), además de métricas de conservación que veremos posteriormente como PhyloP (46). Esta característica es lo que hace que este grupo de métodos sea conocido como *ensemble scores*, ya que usan como *features* las predicciones de otras herramientas junto con las anotaciones clásicas, consiguiendo que la información inherente a estos métodos se incluya también en la predicción global. CADD utiliza un gran conjunto de variantes simuladas para construir su modelo, basado en un algoritmo de *Support Vector Machine* (SVM),

permitiendo además la clasificación de variantes genómicas no codificantes. Este enfoque se escapa del planteamiento de este trabajo, pero es una característica muy interesante que en los últimos años ha tenido grandes resultados a la hora de predecir la posible patogenicidad de zonas que no corresponden a productos génicos, pero que con las últimas investigaciones se ha visto que sí pueden resultar vitales en la función de otras regiones codificantes (47) (5).

- DANN

Este algoritmo desarrollado en 2014 surge para mejorar el rendimiento del anterior clasificador, CADD, cuyo algoritmo interno impedía que se captaran las relaciones no lineales entre variables, algo inherente al *kernel* lineal usado por el *Support Vector Machine*. En este caso los investigadores utilizan una red neuronal profunda (*Deep Neural Network*) para captar todas las posibles relaciones existentes entre las numerosas variables del conjunto de datos, que por otra parte no presenta diferencias respecto al anterior modelo de CADD, ya que utilizan el mismo método de obtención de anotaciones y los mismos datos de entrenamiento (48).

- FATHMM-MKL

Implementado en 2015 como actualización de su predecesor FATHMM, este método de predicción está basado en la obtención de numerosas variables relacionadas con múltiples factores, como el contexto genómico, datos sobre estructura de la cromatina y zonas de unión de factores de transcripción o información procedente de experimentos ChIP-Seq. Utiliza como conjunto de datos para la evaluación de la herramienta variantes del repositorio HGMD y del proyecto 1000 genomas, especialmente aquéllas con una frecuencia alélica superior al 1 %; por otro lado, su algoritmo está basado en un SVM, concretamente un clasificador de tipo *multiple kernel learning* (MKL), de ahí su nombre (49).

- MetaLR

Desarrollado en 2015 por los mismos investigadores que crearon el repositorio dbNSFP, cuya información nos ha servido para obtener las predicciones en este trabajo, y que vieron la necesidad de construir un nuevo modelo que englobara alguno de los predictores existentes hasta el momento. Se trata de un algoritmo basado en regresión logística que utiliza como variables de estudio diferentes predictores, como SIFT, PolyPhen, MutationTaster (50) o PhyloP, además de contar con el factor poblacional mediante la inclusión de la frecuencia alélica máxima de la variante en cuestión. Utilizan para entrenar y calibrar el modelo un conjunto de datos procedente de la base de datos UniProt (51).

- MetaSVM

Mismas características que hemos mencionado anteriormente para el predictor MetaLR, ya que fue desarrollado a la par por dichos investigadores. Solamente difiere del anterior en su algoritmo interno, ya que en este caso MetaSVM depende de un *Support Vector Machine* para captar las relaciones existentes entre las variables ya mencionadas (51).

- REVEL

Predictor desarrollado en 2016, cuya fortaleza está en que presenta la mayor cantidad de clasificadores de patogenicidad formando parte de sus *features*, como son MutPred (52), FATHMM, VEST, PolyPhen, SIFT, PROVEAN (53), MutationAssessor (54), MutationTaster, LRT (55), GERP (56), SiPhy (57), phyloP (46), y phastCons (58). Para obtener los resultados de predicción utiliza como algoritmo un *Random Forest* con hasta 1001 árboles de clasificación, usando conjuntos de variantes procedentes de HGMD y del *Exome Sequencing Project* para entrenar y validar el modelo (59).

- Eigen

Implementado en 2016, este predictor cuenta con una particularidad que lo hace destacar sobre el resto, como es la aplicación de un algoritmo estadístico no supervisado, a diferencia de la gran mayoría de predictores que utilizan un enfoque supervisado (en el que cuentas con dos grupos de valores, etiquetados como una u otra clasificación para que el modelo entrene sobre dichos ejemplos). Utiliza un modelo jerárquico en el que las predicciones de otros métodos se combinan para dar una puntuación final ponderada, a través de la construcción de la matriz de correlaciones y el cálculo de los vectores propios correspondientes a los diferentes predictores. Para su entrenamiento y posterior validación hace uso de un conjunto de variantes del propio repositorio de dbNSFP (60).

- BayesDel

Este predictor, como su propio nombre indica, implementa un algoritmo basado en Naïve Bayes para obtener una clasificación en forma de puntuación numérica mediante la ponderación de numerosos predictores ya existentes, siguiendo el modelo de trabajo de los métodos ensemble. Como principales *features* cuenta con los predictores SIFT, MutationTaster, LRT, FATHMM o PhyloP, entre otros, además del factor poblacional introduciendo frecuencias alélicas en diversos proyectos de secuenciación masiva como los 1000 genomas o ExAC. Utiliza como conjuntos de datos para el testeo de la herramienta variantes procedentes de ClinVar y UniProtKB (61).

- ClinPred

Implementado en 2018, el método de predicción de patogenicidad ClinPred es capaz de obtener las clasificaciones correspondientes utilizando un modelo de *machine learning* basado en dos algoritmos complementarios, *Random Forest* y *Gradient Boosting Decision Tree*, usando como variables explicativas numerosas anotaciones procedentes del repositorio dbNSFP: predictores de patogenicidad existentes (SIFT, PolyPhen-2 HDIV, PolyPhen-2 HVAR, LRT, MutationAssessor, PROVEAN, CADD, GERP, DANN, PhastCons, fitCons (62), PhyloP y SiPhy) y frecuencias alélicas para diferentes poblaciones procedentes del proyecto gnomAD. Tal y como se indica en su nombre, se utilizaron variantes de ClinVar como conjunto de datos para entrenar y calibrar el modelo, catalogadas bien como patógenas o benignas (63).

2.2.3 Predictores de patogenicidad basados en conservación

- GERP++

Este último grupo de predictores de patogenicidad se basan única y exclusivamente en el cálculo de la conservación de la posición específica donde se produce la variante, por lo que la puntuación numérica representa cómo ha cambiado ese nucleótido en las diferentes especies relacionadas filogenéticamente con el humano con el paso del tiempo. GERP++ es uno de estos métodos, desarrollado en 2010, y uno de los más asentados en este tipo de procedimientos de análisis de patogenicidad en entornos clínicos. Se basa en medir mediante alineamientos múltiples la ratio de evolución de cada posición en el genoma, detectando así regiones que se encuentren en limitación selectiva, es decir zonas que no son favorables al cambio y la generación de nuevas variantes neutrales o benignas; por tanto, cuanto mayor sea este factor en una posición específica más potencialmente patógena puede ser la variante que se dé en esta región. GERP++ utiliza para el cálculo de esta métrica los genomas de hasta 46 especies de vertebrados emparentados con el humano, desde los primates más cercanos, construyendo un enorme árbol filogenético con el que visualizar las relaciones entre estas secuencias (56).

- PhyloP

Implementado en 2006, PhyloP es otro predictor de la conservación de nucleótidos en secuencias genómicas mediante el cálculo de la presión selectiva con un algoritmo basado en *Hidden Markov Models*, a diferencia de la estimación de la ratio evolutiva del anterior método. La particularidad de esta herramienta es que no es necesario asumir ninguna estimación evolutiva para hacer los cálculos predictivos, por lo que es un método más generalista. Utiliza al igual que GERP++ secuencias genómicas de una gran cantidad de especies cercanas, hasta 20 mamíferos y 7 vertebrados (46).

Para concluir este apartado hay que hacer una mención especial al repositorio dbNSFP, de donde se han obtenido todas las predicciones de patogenicidad para nuestras variantes. Se trata de un recurso elaborado con la intención de reunir y recopilar en un solo lugar una gran cantidad de predicciones y anotaciones diversas para la gran mayoría de posibles variantes en el genoma humano, a completa disposición para la comunidad científica; para ello los autores han analizado las más de 80.000.000 de variantes no sinónimas y en sitios de *splicing* con todo tipo de anotaciones biológicas, entre las que se incluyen la mayoría de predictores de patogenicidad existentes en la bibliografía, actualizados cada varios meses, y una gran cantidad de frecuencias poblacionales de los principales proyectos de secuenciación masiva. A través de su propia herramienta de búsqueda es posible anotar un conjunto de variantes fácilmente mediante línea de comandos (recibe un fichero *input* con las coordenadas genómicas y los alelos de cada variante y da como *output* una tabla con las diferentes anotaciones detectadas para dichas variantes), además de poder utilizarlo como *plugin* en anotadores estándares como Ensembl-VEP o SnpEff, lo que permite no tener que recorrer todos los servidores web de cada predictor o tener que instalar en local los códigos fuente, con todos los recursos temporales y competencias informáticas que conlleva (51).

2.3 Conjuntos de datos

El siguiente paso en el flujo de trabajo se corresponde con la elección de los conjuntos de variantes adecuados para llevar a cabo la evaluación de los predictores descritos anteriormente, para lo cual es preciso indagar e investigar en las numerosas revisiones y artículos publicados sobre el tema, para ver qué conjuntos de datos se han ido utilizando a lo largo del tiempo, qué resultados se han obtenido gracias a ellos y cuáles podrían ser los más apropiados. En la siguiente Tabla 2 podemos observar algunos de los conjuntos de variantes que más se han utilizado en la bibliografía revisada, incluyendo revisiones y *benchmarkings* de diferentes predictores como los propios artículos donde se describe la implementación de cada método, en el que se usan datos para entrenar y evaluar el modelo (12–16,64,65).

Tabla 2. Lista con diferentes conjuntos de variantes usados para construir y revisar modelos de predicción de patogenicidad.

Nombre	Enlace web	Comentarios
ClinVar	https://www.ncbi.nlm.nih.gov/clinvar/ (22)	Clasificación de patogenicidad basado en evidencia
TP53 IARC	https://p53.iarc.fr/ (66)	Variantes de carácter tumoral en el gen TP53
ICGC	https://dcc.icgc.org/ (67)	Repositorio de variantes detectadas en numerosos tipos de cánceres
dbSNP	https://www.ncbi.nlm.nih.gov/snp/ (68)	Repositorio principal de variantes de un único nucleótido
VariBench	http://structure.bmc.lu.se/VariBench/ (69)	Conjuntos de datos usados para el entrenamiento de modelos de predicción de patogenicidad
UniProt	https://www.uniprot.org/ (70)	Repositorio principal de secuencias proteicas
1000 genomes	https://www.internationalgenome.org/data-portal/sample (71)	Genomas secuenciados de numerosas subpoblaciones
gnomAD/ExAC	https://gnomad.broadinstitute.org/ (72)	Genomas y exomas secuenciados de casi 150.000 participantes
HGMD	http://www.hgmd.cf.ac.uk/ac/index.php (73)	Base de datos público/privada de variantes con información clínica
PhenCode	http://phencode.bx.psu.edu/ (74)	Recopilatorio de variantes con información fenotípica mantenida por la UCSC (75).
DoCM	http://docm.info/ (76)	Repositorio de variantes causantes de enfermedad y revisadas para asegurar una alta calidad
GIAB	https://www.nist.gov/programs-projects/genome-bottle (77)	Proyecto de obtención de conjuntos de variantes gold standard

Las características de estos repositorios varían entre uno y otro: ciertos proyectos de secuenciación masivos, como el de los 1000 genomas o el gnomAD, son la mayor fuente existente de variantes genéticas humanas, con información muy completa sobre frecuencias poblacionales, mientras que otras bases de datos más enfocadas en el ámbito clínico contienen conjuntos de datos con información sobre fenotipos y su relación con diversas enfermedades, entre ellos ClinVar o HGMD. Por tanto, vemos como la elección de uno u otro conjunto de datos puede ser vital en este sentido, pues el enfoque de nuestro trabajo, ya sea una evaluación estadística o la propia construcción de un modelo de predicción, se verá afectado por el propio sesgo o las características inherentes a las variantes de estudio.

Para nuestro caso concreto, al tratarse de un proyecto de evaluación de diferentes métodos de priorización clínica de variantes necesitaremos la mayor exhaustividad posible en relación a los conjuntos de datos, intentando captar al máximo todos los posibles puntos que requiere un buen clasificador de patogenicidad para poder ser implementado adecuadamente en la práctica clínica. A continuación, procedemos a enumerar y describir brevemente los diferentes *datasets* que hemos elegido para este fin:

2.3.1 Variantes patógenas y benignas de ClinVar

El repositorio ClinVar, perteneciente al NCBI, es la principal fuente de información existente acerca de la relación entre variantes genéticas humanas y fenotipos clínicos, con multitud de variantes descritas por la comunidad científica y etiquetadas como patógenas o benignas según un sistema de clasificación basado en 5 niveles: *pathogenic*, *likely pathogenic*, *uncertain significance*, *likely benign* y *benign*. Es la base de datos más usada por los predictores de patogenicidad desarrollados hasta la fecha, ya que su información está organizada en niveles de evidencia que aseguran con casi total certeza la clasificación de una variante determinada como patógena o benigna.

En nuestro trabajo hemos recogido el fichero en formato VCF (78) de todas las variantes descritas en ClinVar a fecha de 18/09/2020, modificado para quedarnos sólo con la información que nos interesa (cromosoma, posición genómica, alelo de referencia y alelo alternativo); posteriormente llevamos a cabo una serie de filtros para seleccionar aquellas variantes más indicadas para el estudio: las catalogadas con mínimo una clasificación de 2 estrellas o más (*criteria_provided,_multiple_submitters,_no_conflicts*, *practice_guideline* o *reviewed_by_expert_panel*), las *missense Single Nucleotide Variants* (SNVs no sinónimos, que provocan un cambio de aminoácido en la secuencia proteica) y las etiquetadas como *pathogenic*, *likely pathogenic*, *likely benign* o *benign* (separadas dos a dos en sendos ficheros, uno para las variantes potencialmente patógenas y otro para las benignas, que más adelante serán unidos para su evaluación). Al final se obtienen un total de 5515 variantes patógenas y 10896 variantes benignas para elaborar los informes de evaluación en R.

2.3.2 Variantes somáticas de TP53 e ICGC

Para contar con un enfoque diferente al de evaluar variantes relativamente comunes y de carácter germinal decidimos obtener un segundo conjunto de datos con variantes somáticas procedentes de los repositorios IARC (66,79,80) e ICGC (67), con los que se recogieron datos sobre variantes tumorales en el gen TP53 (uno de los genes más importantes en procesos cancerígenos y ampliamente estudiado en términos de polimorfismos genéticos desencadenantes del tumor) y de diversas variantes somáticas detectadas en muestras tumorales, respectivamente. Nuestro objetivo al recoger estos datos

es ofrecer una evaluación de cómo se comportan los diferentes predictores para clasificar este tipo de variantes según su patogenicidad, puesto que se trata de uno de los grandes retos en la investigación oncológica.

Este conjunto de variantes se obtuvo a partir de los datos ya puestos a punto en el trabajo *Performance evaluation of pathogenicity-computation methods for missense variants* (14), que ya hizo un enfoque parecido y donde ya se filtraron las variantes según nuestras especificaciones (*missense* SNVs); para completar el trabajo llevamos a cabo la división entre variantes patógenas y benignas en dos ficheros diferentes para su posterior estudio en R, obteniendo al final un total de 1133 variantes patógenas y 590 variantes benignas.

2.2.3 Variantes de la muestra NA12878 del proyecto GIAB (Genome in a Bottle)

Como último conjunto de datos para llevar a cabo nuestra evaluación escogimos el fichero VCF de todas las variantes codificantes detectadas en la muestra NA12878, uno de los individuos englobados dentro del consorcio de *Genome in a Bottle* (24). Esta organización tenía como objetivo en su creación el de secuenciar un gran número de muestras humanas y de aportarlas en abierto a la comunidad científica caracterizadas como *gold standars*, esto es, conjuntos de variantes que con bastante certeza se corresponden con las variantes reales del individuo (con el menor número de falsos positivos o variantes falsas) y que sirven como punto de referencia para todo tipo de algoritmos y herramientas de detección y clasificación de variantes.

En este caso, nuestro objetivo con el conjunto de variantes de GIAB no es el de evaluar métricas de clasificación, puesto que estos datos no cuentan con información clínica que permitan servir de referencia para su comparación con las predicciones, sino de comprobar con un caso real cómo se comportan los diferentes métodos de predicción y cómo concuerdan sus predicciones, asentando las bases para su implementación en la práctica clínica. En total contamos con unas 10342 variantes de tipo SNV y *missense*, al igual que en los anteriores conjuntos de datos.

2.4 Métricas estadísticas

Para evaluar el rendimiento de cualquier algoritmo clasificador es común hacer uso de determinadas métricas estadísticas que nos permiten medir los resultados del predictor y compararlos con los demás, estableciendo un marco claro y objetivo para diferenciar clasificadores y escoger el que mejores resultados nos ofrece. El caso de las predicciones de patogenicidad en variantes genéticas no es una excepción, por lo que también aplicaremos todo tipo de cálculos y métricas para analizar y describir de forma clara el rendimiento de los predictores escogidos. Por tanto, en este punto pasaremos a enumerar y explicar de forma breve las métricas que utilizaremos en el análisis, su fórmula matemática, sus propiedades específicas y el entorno de trabajo en el que llevaremos a cabo su cálculo (16,81).

Para comenzar hay que tener en cuenta que este proyecto se realizará mediante el lenguaje de programación R y su entorno de desarrollo RStudio, por lo que haremos uso de paquetes específicos de este lenguaje para calcular todas las métricas que describiremos a continuación. El primer paso es conocer y construir la llamada matriz de confusión, a partir de la cual podemos ir obteniendo cada una de las métricas mediante combinaciones de los diferentes elementos de la matriz. Esta tabla representa la cantidad de predicciones correctas e incorrectas para un clasificador binario, esto es, un predictor con sólo dos posibles resultados, 1/0 o positivo/negativo; en nuestro caso las predicciones positivas se corresponden con la etiqueta '*pathogenic*/*deleterious*', codificada con un 1, mientras que las predicciones negativas se refieren a la etiqueta '*benign*/*neutral*', codificada con un 0. Los valores que se muestran en la tabla son los *True Positives* (TP), *False Positives* (FP), *True Negatives* (TN) y *False Negatives* (FN), siendo los *True* las predicciones correctas y los *False* las incorrectas, por lo que con un simple vistazo podemos ver cómo de bien ha ido la predicción de la patogenicidad de las variantes. En función de estos valores se calculan las consiguientes métricas, que pasaremos a enumerar a continuación en la Tabla 3.

Tabla 3. Lista de métricas estadísticas utilizadas en la evaluación de las predicciones de patogenicidad.

Métrica	Fórmula	Método/paquete de cálculo
Análisis de concordance	-	*Comparar dos a dos todos los métodos y calcular el porcentaje de predicciones iguales.*
AUC ROC (Area Under Receiver Operating Characteristic)	-	*Paquete ROCit de R*
PPV (Positive Predictive Value, Precision)	$Precision = \frac{TP}{TP + FP}$	*Paquete caret de R*
NPV (Negative Predictive Value)	$NPV = \frac{TN}{TN + FN}$	*Paquete caret de R*
Sensitivity	$Sensitivity = \frac{TP}{TP + FN}$	*Paquete caret de R*
Specificity	$Specificity = \frac{TN}{TN + FP}$	*Paquete caret de R*
Accuracy	$Accuracy = \frac{TP + TN}{TN + FN + TP + FP}$	*Paquete caret de R*
MCC (Matthews Correlation Coefficient)	$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP \times FN) \times (TP \times FP) \times (TN \times FP) \times (TN \times FN)}}$	*Paquete mcc de R*
F1 score	$F1 = 2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity}$	*Paquete caret de R*
kappa	$k = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)}$	*Paquete caret de R*
Prevalence	$Prevalence = \frac{TP + FN}{TP + TN + FP + FN}$	*Paquete caret de R*
Detection rate	$Detection\ rate = \frac{TP}{TP + TN + FP + FN}$	*Paquete caret de R*

Detection Prevalence	$Detection\ prevalence = \frac{TP + FP}{TP + TN + FP + FN}$	*Paquete caret de R*
Balanced Accuracy	$Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2}$	*Paquete caret de R*

La única métrica que no está calculada directamente con los valores de TP, TN, FP y FN es el estadístico *Kappa*, que mide en un rango de valores de 0 a 1 como es la concordancia o el acuerdo existente entre predictores, calculado como la relación existente entre la concordancia real y la esperada en función de los datos. Este tipo de métrica sirve especialmente para analizar los resultados en muestras muy desbalanceadas, en la que uno de los grupos es mayoritario y por tanto puede introducir un sesgo importante en otro tipo de métricas.

Generalmente los clasificadores se evalúan teniendo en cuenta todos estos factores, pero las métricas más importantes y sobre las que se puede captar la idea más completa del predictor son el valor AUC y la *Accuracy*, ya que representan de forma fehaciente todas las posibles propiedades y preferencias del algoritmo. De hecho, en el informe de resultados de R Markdown mostraremos la tabla completa con todas las métricas anteriores para cada uno de los predictores, ordenándolos por el valor AUC para tener una idea a primera vista de cuáles pueden ser los mejores en función de su rendimiento.
Además de todas estas métricas construiremos las curvas ROC para cada método de predicción mediante el paquete ROCit de R, mostrando cada uno de un color para que se distingan adecuadamente y podamos apreciar los resultados de forma gráfica. Esta curva, denominada en inglés *Receiver Operating Characteristic*, representa mediante diferentes valores la relación existente entre la *Sensitivity* y el factor 1 - *Specificity*, ofreciendo una imagen muy completa de cómo se comporta el clasificador; cuanto más se acerque la curva al extremo superior izquierdo, que representa el 100 % de estos dos valores, mejor será el resultado de un clasificador, mientras que cuanto más se acerque a la diagonal peores serán los resultados obtenidos, ya que esta zona representaría el clasificador al azar (que elige sin razón y obtiene probabilidades de acierto y error de 50 y 50 %, respectivamente).

2.5 Plataformas de priorización

En esta primera parte del segundo bloque del proyecto, donde se analizarán diferentes plataformas de interpretación y priorización de variantes, se recogen y describen aquellos métodos que vamos a evaluar, explicando brevemente sus características y el tipo de anotación e información que nos aporta, además de su potencial uso como herramienta en la práctica clínica.

Las diferentes plataformas de interpretación de variantes existentes se basan en varios factores que permiten su comparación: son normalmente servidores web donde el usuario puede subir sus ficheros de variantes en formatos como VCF o CSV, estándares en el almacenamiento de variantes genéticas, para obtener los resultados esperados; ofrecen una serie de anotaciones como variables de salida, normalmente en formato de columnas con la opción de exportar a un fichero de salida tabulado, que permiten caracterizar a las variantes introducidas, como por ejemplo información estructural y funcional a

nivel de variante (posición genómica, situación dentro del gen y/o proteína, consecuencia funcional, frecuencia alélica en diferentes subpoblaciones, etc.) o a nivel de gen (tipo de proteína, términos ontológicos de función biológica, proceso celular, enfermedades relacionadas, etc); suelen ofrecer avisos y notificaciones a una dirección de correo electrónico para darte directamente los resultados a descargar o el enlace con los que visualizarlos en su propia plataforma; o, por último, dan la posibilidad de construir y realizar filtros algorítmicos, más o menos complejos, con los que seleccionar y priorizar manualmente los resultados que mejor nos convengan, como introducir regiones génicas o fenotipos con los que filtrar la lista de genes o variantes que aparezcan finalmente.

A continuación vamos a ir presentando y describiendo brevemente las diferentes plataformas de priorización escogidas, como son VarCards (82), GenIO (83), MutationDistiller (84) y OpenCRAVAT (85), en función de sus características principales y sus potenciales usos y aplicaciones.

- VarCards

Esta herramienta de anotación de variantes, desarrollada en 2017, ofrece un entorno muy sencillo para aquellos usuarios sin habilidades bioinformáticas para la anotación y priorización de variantes genéticas, todo ello a través de una plataforma web. No requiere ningún tipo de registro, solamente introducir las variantes en el formato adecuado, ya sea en VCF o en el basado en la herramienta ANNOVAR (86), y marcar las diferentes opciones que se nos ofrece para anotar las variantes con las especificaciones que mejor se adapte a nuestro enfoque. El conjunto de anotaciones es bastante completo en general, teniendo siempre en cuenta el estándar en estos casos aportado por Ensembl-VEP (87), incluyendo desde información básica a nivel de variante y gen, coordenadas y enlaces a diferentes repositorios, predicciones computacionales de patogenicidad de todo tipo y frecuencias poblacionales de las principales bases de datos. En este sentido, la información aportada no supera en gran medida la aportada por otras plataformas estándares de anotación, pero su rápida descarga en forma Excel y un formato limpio y adecuado de los campos de salida hacen que sea una alternativa bastante interesante.

- GenIO

La plataforma de priorización GenIO, implementada también en este caso en 2018, representa un paso más allá a la mera anotación de variantes y ofrece un nuevo enfoque donde el proceso de priorización ya no se deja completamente a responsabilidad del usuario, sino que asiste directamente en el filtro y selección de variantes más importantes mediante una lista priorizada según una serie de factores, como son las clasificaciones basadas en ClinVar (88), InterVar (interpretador clínico mediante las guías prácticas de la ACMG) (89), M-CAP (clasificador de patogenicidad especialmente dirigido a variantes raras de tipo *missense*) (90) y el potencial impacto en la proteína obtenido mediante Ensembl. No obstante, la particularidad esencial de este tipo de plataformas de priorización, y que las diferencia de las propias herramientas básicas de anotación, es que permite la introducción de términos médicos para filtrar y optimizar el proceso de priorización de variantes, como son los síntomas observados, los términos estandarizados de enfermedades y condiciones

genéticas o las observaciones complementarias establecidas por un clínico, por lo que en este caso la lista ofrecida está enfocada en los términos fenotípicos que hayamos introducido. Además de esto se obtiene también una lista de las variantes introducidas anotadas por múltiples fuentes de datos, con la opción de descargarlo en formato VCF.

- MutationDistiller

Desarrollada en 2019, esta plataforma permite la priorización de variantes genéticas contenidas en un fichero en formato VCF mediante la obtención de una lista ordenada de potenciales variantes causales de alguna patología, por lo que se trata de la única herramienta analizada en este trabajo que es capaz de ofrecer un orden de prioridad sobre las variantes introducidas. Para el análisis sólo necesita la lista de variantes en formato VCF y la selección de determinados parámetros que refinarán o filtrarán la búsqueda en función de nuestros intereses: regiones génicas a analizar, términos fenotípicos en diferentes formatos (HPO, Orphanet, OMIM), rutas metabólicas involucradas en el fenotipo que estamos buscando o el modo de herencia, toda la información que la herramienta tiene en cuenta para realizar la priorización. Como resultados obtenemos la lista de variantes ordenadas en función de un score que depende de varios factores, algunos de los cuales introducidos por el usuario como el modo de herencia o la condición fenotípica en cuestión, pero otros inherentes a la plataforma como el predictor de patogenicidad MutationTaster (91), sobre el que se obtiene la probabilidad de que una variante determinada sea potencialmente patógena. En cuanto a la información ofrecida no vemos una gran cantidad de las llamadas anotaciones básicas como en el resto de plataformas, sino que en este caso MutationDistiller se centra en la relación de la variante con el fenotipo, presentando información según las bases de datos de enfermedades genéticas como OMIM u Orphanet, además de datos sobre frecuencias poblacionales; sin embargo, no ofrece ningún tipo de dato sobre diferentes predictores de patogenicidad *in silico*.

- OpenCRAVAT

La herramienta de anotación y priorización OpenCRAVAT, implementada justo este mismo año 2020, trata desde un enfoque completamente integral el proceso de anotación de variantes genéticas, llegando a obtener como resultado una serie de tablas y gráficos con hasta 156 tipos de anotaciones, desde la información genómica más básica, pasando por toda clase predictores de patogenicidad y frecuencias poblacionales en la mayor parte de repositorios existentes en la actualidad hasta métricas de conservación o anotaciones muy específicas para determinadas enfermedades o casos concretos, como por ejemplo el predictores de patogenicidad de variantes somáticas/tumorales CHASM (92). Todo esto hace que cuantitativamente a nivel de información ofrecida supere al resto de herramientas analizadas anteriormente, por lo que resulta una plataforma muy completa y exhaustiva para anotar y priorizar un conjunto de variantes. Como contrapartida no posee la capacidad de priorizar mediante algún tipo de lista las variantes analizadas por su potencial causa de algún síndrome o enfermedad, tal y como llevan a cabo GenIO y MutationDistiller, sino que se obtienen los resultados en forma de diferentes gráficos y tablas que podemos descargar para contar con ellos en local. La presencia de gráficos visualmente atractivos es una particularidad que hace

que OpenCRAVAT destaque también frente a las demás, permitiendo que se puedan sacar conclusiones rápidas sobre el conjunto de variantes mediante un rápido vistazo en los numerosos gráficos que ofrece.

3. Resultados y discusión

3.1 Evaluación métodos de predicción de patogenicidad

En este primer apartado de resultados se describirán algunas de las tablas y gráficos obtenidos en los informes R Markdown donde se llevaron a cabo los análisis, adjuntos junto con esta memoria en la parte de Anexos, además de discutir e intentar comprender el significado de éstos en el contexto de la interpretación de la patogenicidad de variantes. Se realizó el mismo tipo de análisis de evaluación para los tres conjuntos de datos que hemos descrito previamente, a saber: cálculo de métricas estadísticas de evaluación de clasificadores, curva ROC y área asociada para cada predictor (AUC) y análisis de concordancia entre métodos, para lo cual se construirán *heatmaps* o mapas de calor; excepcionalmente, el conjunto de variantes de GIAB sólo contará con el análisis de concordancia, puesto que al no contar con información clínica asociada no podemos evaluar las predicciones con los valores reales.

En primer lugar, se mostrarán los cálculos detallados de cada métrica para cada uno de los predictores en los conjuntos de datos de ClinVar y de IARC/ICGC en las tablas 4 y 5, respectivamente.

Tabla 4. Métricas estadísticas para evaluar las predicciones de los diferentes métodos para el conjunto de variantes de ClinVar ordenados según su valor de AUC.

	Sensitivity	Specificity	Neg.Pred. Value	Precision	F1	Prevalence	Detection. Rate	Detection. Prevalence	Balanced. Accuracy	Accuracy	Kappa	MCC	AUC
ClinPred	0.9318389	0.9874504	0.9662302	0.9740903	0.9524963	0.3361309	0.3132198	0.3215511	0.9596446	0.9687577	0.9292384	0.9297455	0.9596446
BayesDel_	0.9372620	0.9702370	0.9682802	0.9410158	0.9391352	0.3362600	0.3151637	0.3349186	0.9537495	0.9591488	0.9083930	0.9083971	0.9537495
REVEL	0.9632825	0.8289388	0.9781659	0.7394307	0.8366420	0.3350763	0.3227731	0.4365157	0.8961107	0.8739542	0.7368893	0.7539863	0.8961107
MetaSVM	0.8836057	0.8806550	0.9375554	0.7886285	0.8334199	0.3350763	0.2960753	0.3754306	0.8821303	0.8816437	0.7420932	0.7449791	0.8821303
VEST4	0.9704153	0.7838686	0.9813980	0.6927719	0.8084194	0.3343163	0.3244256	0.4683008	0.8771419	0.8462342	0.6858690	0.7131027	0.8771419
MetaLR	0.8762622	0.8594690	0.9323565	0.7585823	0.8131868	0.3350763	0.2936147	0.3870571	0.8678656	0.8650960	0.7084707	0.7129833	0.8678656
DEOGEN2	0.8404481	0.8559535	0.9117427	0.7518576	0.7936884	0.3418064	0.2872706	0.3820811	0.8482008	0.8506536	0.6772229	0.6798031	0.8482008
Eigen	0.9428848	0.6754241	0.9593776	0.5926016	0.7277890	0.3336563	0.3145995	0.5308786	0.8091545	0.7646641	0.5388038	0.5842034	0.8091545
SIFT	0.9104562	0.6606923	0.9359215	0.5754627	0.7051981	0.3356225	0.3055696	0.5309980	0.7855742	0.7445186	0.4992435	0.5404409	0.7855742
LIST.S2	0.8803675	0.6807762	0.9181527	0.5831574	0.7015840	0.3365518	0.2962893	0.5080777	0.7805719	0.7479490	0.4985463	0.5303838	0.7805719
CADD_hg19	0.9700816	0.5688326	0.9740688	0.5324443	0.6875281	0.3360551	0.3260009	0.6122723	0.7694571	0.7036744	0.4479909	0.5224625	0.7694571
DANN	0.8759746	0.6376652	0.9103774	0.5502905	0.6759480	0.3360551	0.2943757	0.5349461	0.7568199	0.7177503	0.4481473	0.4864333	0.7568199
phyloP100way	0.9040798	0.5485499	0.9186904	0.5033821	0.6466926	0.3360551	0.3038206	0.6035586	0.7263149	0.6680275	0.3782773	0.4370842	0.7263149
fathmm.MKL	0.9735267	0.4469530	0.9708931	0.4711716	0.6350089	0.3360551	0.3271586	0.6943513	0.7102399	0.6239108	0.3328509	0.4311372	0.7102399
PrimateAI	0.3757391	0.9637345	0.7566724	0.8372291	0.5186940	0.3317515	0.1246520	0.1488864	0.6697368	0.7686662	0.3941782	0.4490143	0.6697368
GERP	0.9577516	0.3356290	0.9400720	0.4219861	0.5858474	0.3361780	0.3219750	0.7629991	0.6466903	0.5447729	0.2233874	0.3259154	0.6466903

Tabla 5. Métricas estadísticas para evaluar las predicciones de los diferentes métodos para el conjunto de variantes somáticas de IARC/ICGC ordenados según su valor de AUC.

	Sensitivity	Specificity	Neg.Pred. Value	Precision	F1	Prevalence	Detection. Rate	Detection. Prevalence	Balanced. Accuracy	Accuracy	Kappa	MCC	AUC
VEST4_pred	0.9551282	0.6847458	0.8918322	0.8486574	0.8987505	0.6492271	0.6200951	0.7306778	0.8199370	0.8602854	0.6759514	0.6883459	0.8199370
Eigen_pred	0.9302536	0.6615120	0.8333333	0.8390523	0.8823024	0.6548043	0.6091340	0.7259786	0.7958828	0.8374852	0.6220894	0.6307890	0.7958828
CADD_hg19_pred	0.9576346	0.5779661	0.8766067	0.8133433	0.8796109	0.6575740	0.6297156	0.7742310	0.7678004	0.8276262	0.5832120	0.6078961	0.7678004
ClinPred_pred	0.9628975	0.5372881	0.8830084	0.7997065	0.8737475	0.6573751	0.6329849	0.7915215	0.7500928	0.8170732	0.5519174	0.5843665	0.7500928
DEOGEN2_pred	0.8532853	0.6307692	0.6936090	0.8144330	0.8334066	0.6550708	0.5589623	0.6863208	0.7420273	0.7765330	0.4946635	0.4959033	0.7420273
phyloP100way	0.8729038	0.6033898	0.7120000	0.8086672	0.8395586	0.6575740	0.5739988	0.7098085	0.7381468	0.7806152	0.4943644	0.4979864	0.7381468
DANN_pred	0.8314210	0.6423729	0.6649123	0.8169991	0.8241470	0.6575740	0.5467208	0.6691817	0.7368969	0.7666860	0.4776740	0.4778354	0.7368969
SIFT_pred	0.9141274	0.5153584	0.7645570	0.7770801	0.8400509	0.6488916	0.5931696	0.7633313	0.7147429	0.7741162	0.4642027	0.4823126	0.7147429
PrimateAI_pred	0.4000000	0.9910714	0.4556650	0.9888143	0.5695876	0.6636637	0.2654655	0.2684685	0.6955357	0.5987988	0.3032120	0.4169211	0.6955357
fathmm.MKL	0.9496911	0.4084746	0.8087248	0.7550877	0.8412823	0.6575740	0.6244922	0.8270459	0.6790828	0.7643645	0.4063582	0.4493755	0.6790828
GERP_pred	0.9311562	0.4129693	0.7562500	0.7541101	0.8333333	0.6591041	0.6137289	0.8138453	0.6720628	0.7545084	0.3864733	0.4190798	0.6720628
LIST.S2_pred	0.8498635	0.4829932	0.6325167	0.7544426	0.7993154	0.6514523	0.5536455	0.7338471	0.6664284	0.7219917	0.3522137	0.3588900	0.6664284
REVEL_pred	0.9276786	0.3497453	0.7177700	0.7306610	0.8174666	0.6553540	0.6079579	0.8320655	0.6387120	0.7284962	0.3158090	0.3527116	0.6387120
BayesDel_addAF_pred	0.9390997	0.2372881	0.6698565	0.7027741	0.8039290	0.6575740	0.6175276	0.8786999	0.5881939	0.6987812	0.2086797	0.2563738	0.5881939
MetaSVM_pred	0.8107143	0.1103565	0.2346570	0.6340782	0.7115987	0.6553540	0.5313049	0.8379169	0.4605354	0.5693388	0.0902636	0.1017871	0.4605354
MetaLR_pred	0.8008929	0.1018676	0.2120141	0.6290323	0.7046347	0.6553540	0.5248683	0.8344061	0.4513802	0.5599766	0.1108971	0.1243245	0.4513802

En estas dos tablas se pueden ver todas las métricas calculadas para cada uno de los predictores, ordenados según su valor del área bajo la curva ROC (AUC), el último de los campos. En primer lugar hay que comentar que el comportamiento de ciertas métricas sigue un patrón similar, ya que se basan en los mismos valores pero con una perspectiva distinta. De esta forma, métricas cuyo cálculo depende de las cuatro variables principales (TP, TN, FP y FN) se parece en cuanto al orden que presentan para los predictores, como son los valores de AUC, MCC, *Kappa*, *Accuracy* (y *Balanced Accuracy*) o F1, puesto que su valor descendente es prácticamente similar para todos los predictores; sin embargo, otras métricas que dependen de sólo algunas variables no siguen este patrón, por lo que nos servirán para identificar y caracterizar comportamientos individualizados. Entre estos últimos están *Sensitivity*, *Specificity*, *Negative Predictive Value*, *Precision*, *Prevalence* o *Detection Rate* y *Detection Prevalence*. Hay que tener en cuenta que se ha usado como clasificación positiva, es decir como *True Positive*, los valores patógenos, caracterizados en nuestro código de R Markdown como 1s, mientras que las etiquetas benignas se corresponderían con los *True Negatives* o clasificación negativa, descrita como 0 en nuestro código.

Si nos fijamos en la primera tabla para las variantes de ClinVar, que se presupone tienen un carácter germinal y están más estudiadas, los resultados son bastante buenos para una buena parte de los predictores. Sobre todos destacan con valores de AUC de alrededor de 0’95 ClinPred y BayesDel, los dos predictores de patogenicidad más novedosos de los que contamos en la lista, por lo que tiene lógica pensar que son los que cuentan con un conjunto de datos de entrenamiento más actualizado y revisado, con la mejora en la predicción que ello supone. Sin duda el factor diferencial respecto a estos dos predictores es en la *Specificity*, o proporción de *True Negatives* detectados respecto al total de negativos, con valores de 0’987 para ClinPred y 0’970 para BayesDel, que los hacen destacar en este aspecto sobre el resto de predictores. Sin embargo, para la *Sensitivity*, o proporción de *True Positives* respecto al total de positivos, existen otros métodos que los superan por poco margen, como fathmmMKL con un valor de 0’973 y CADD y VEST4 con 0’970, por lo que podríamos considerar a estos predictores como los más sensibles o los que son capaces de detectar mejor la patogenicidad que el carácter benigno de las variantes. Para finalizar con la descripción de las métricas para las variantes de ClinVar es preciso hacer hincapié en ciertos predictores que poseen un comportamiento o patrón extraño, y que mediante esta evaluación pueden salir a la luz. Por ejemplo, phyloP, fathmmMKL y GERP poseen valores muy buenos de *Sensitivity*, por lo que son capaces de encontrar bien los patrones de patogenicidad, pero sin embargo cuentan con valores de detección de negativos muy baja, sobrepasando incluso la barrera del 0’50, lo que hace que sus valores generales como el AUC se resientan y bajen hasta quedar al final de la tabla. Por otro lado, sorprendentemente PrimateAI presenta un comportamiento contrario y diferente respecto al resto de predictores, con un valor de *Specificity* muy alto (0’963) y *Sensitivity* bajo (0’375), lo que nos dice que este predictor detecta mejor las variantes benignas o neutrales que las patógenas.

En la segunda tabla podemos observar el comportamiento de los diferentes predictores respecto al conjunto de variantes somáticas, y lo primero que nos llama la atención es que son en general valores más bajos, sin llegar al menos a un 0’90 de AUC, lo que podríamos esperar conociendo el carácter tumoral de las variantes analizadas. En este caso los predictores con mejores resultados son VEST4, con un valor de AUC de 0’819, y Eigen, con 0’795, aunque ClinPred no se queda muy atrás y presenta un valor de AUC de 0’750, por lo que observamos que es uno de los predictores que mejores resultados está dando. En cuanto al patrón general se observa que se sigue un comportamiento similar que para las variantes de ClinVar, ya que la mayoría de predictores tienen valores de *Sensitivity* generalmente más altos que de *Specificity*, reforzando aún más la idea de que este tipo de algoritmos detectan de forma más precisa las variantes patógenas que las benignas, con los que se obtienen un mayor número de falsos positivos. Sorprende también el caso de PrimateAI, que de nuevo presenta unos valores mucho más altos de *Specificity* y por tanto una mayor capacidad de detectar correctamente las variantes neutras. No obstante, las peores métricas las presentan los predictores MetaSVM y MetaLR, desarrollados por el mismo grupo de investigadores que construyeron el repositorio dbNSFP, que presentan valores de AUC muy por debajo de lo que se considera el umbral mínimo de cualquier algoritmo predictor, 0’50; esto se debe a los valores tan bajos que presentan de *Specificity*, de 0’110 para MetaSVM y 0’101 para MetaLR, una detección tan pobre que debe tener alguna explicación plausible más allá de su mayor o menor precisión a la hora de detectar la patogenicidad de una variante.

Posteriormente se obtuvieron las curvas ROC para cada uno de los predictores, señalando además el valor de AUC para cada uno de ellos, donde podemos hacernos una idea rápidamente de cómo se comporta cada uno de ellos, en las Figuras 2 y 3 para el conjunto de variantes de ClinVar y de IARC/ICGC, respectivamente. Se aprecia perfectamente como para el primer conjunto de datos los predictores con mejor valor AUC son ClinPred y BayesDel, con GERP y PrimateAI siendo los dos que presentan un valor más bajo. Se observa además como para la gran mayoría de predictores la forma de la curva es similar, indicando que siguen un mismo patrón de predicción de patogenicidad, con valores de *Specificity* por debajo de los de *Sensitivity*; sin embargo, PrimateAI tiene una forma completamente distinta, que nos hace ver de nuevo su extraño comportamiento gracias a su alto valor de detección de *True Negatives*.

En la siguiente figura relativa a las variantes somáticas se observan los mismos resultados que hemos comentado previamente con la tabla, donde destacan PrimateAI, de nuevo con su enfoque más sesgado hacia las variantes benignas, y MetaSVM y MetaLR, que aparecen por debajo de la recta con valor 0’50, lo que se considera como un clasificador que predice al azar.

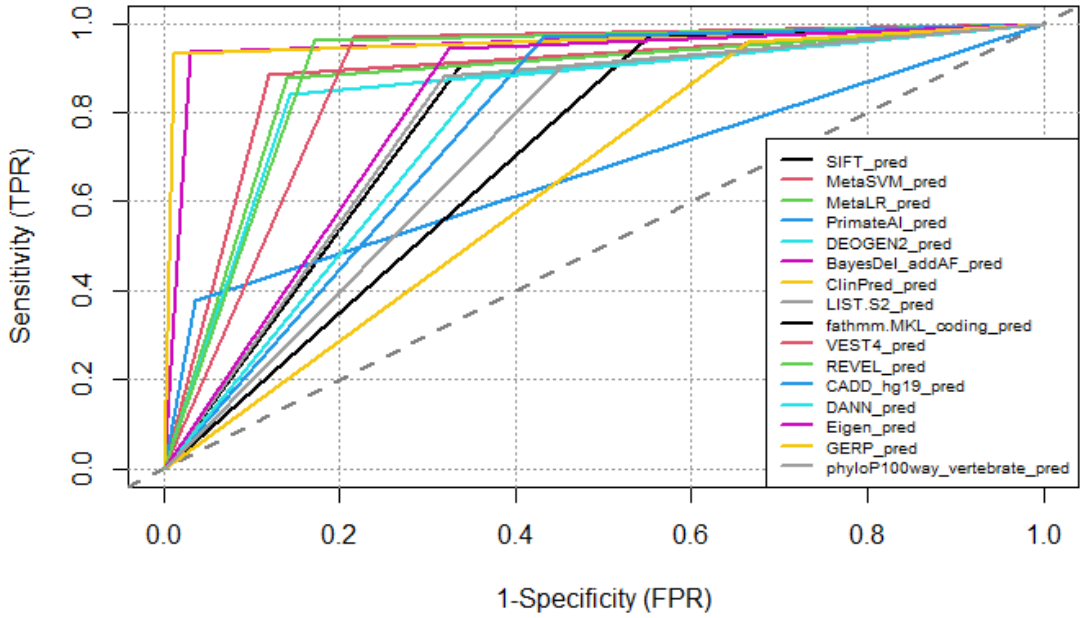


Figura 2. AUC para los diferentes predictores de patogenicidad en el conjunto de variantes de ClinVar.

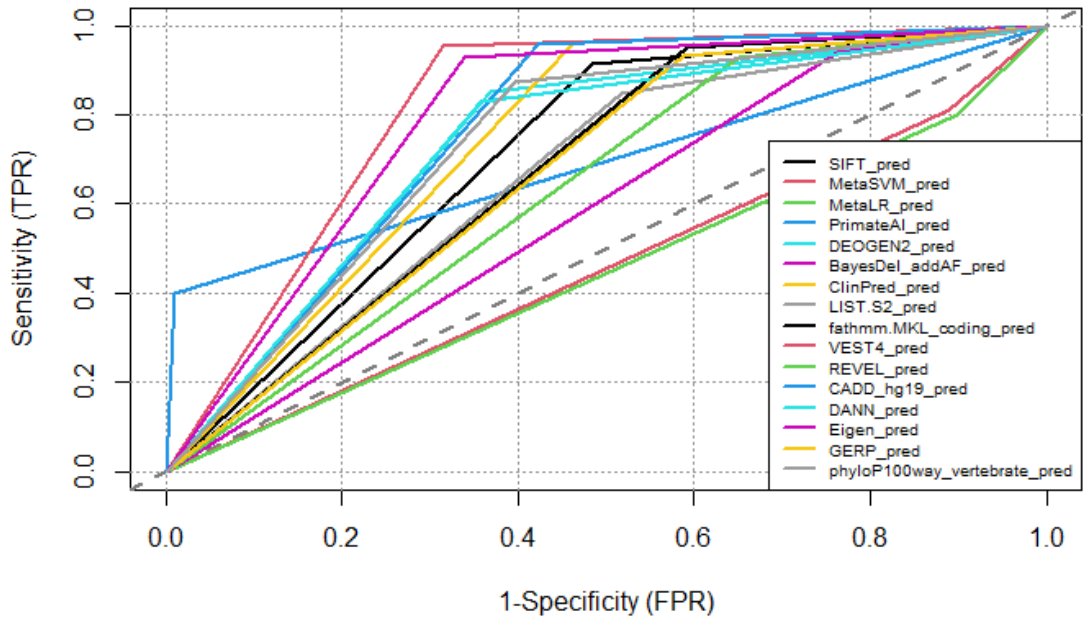


Figura 3. AUC para los diferentes predictores de patogenicidad en el conjunto de variantes somáticas de IARC/ICGC.

Seguidamente se llevó a cabo el análisis de concordancia entre predictores, esto es, la similitud en cuanto a las clasificaciones de los diferentes métodos respecto a cada variante. Este tipo de procedimientos es común en evaluaciones y *benchmarkings*, especialmente cuando se trata de un enfoque más dirigido a la práctica clínica que a la puramente investigadora, puesto que generalmente se requiere que para considerar una variante como patógena o benigna exista un mínimo de predictores que clasifiquen dicha variante en una de las categorías, sin que exista un alto grado de discordancia, que es normalmente lo que sucede en las variantes de significado incierto (VUS). Por tanto, este análisis pretende analizar qué predictores poseen mejores concordancias mediante la construcción de un *heatmap*, donde se aprecia visualmente las predicciones para cada una de las variantes, y un cálculo del porcentaje de variantes que presentan exactamente la misma clasificación, tanto de forma completa como por pares de métodos.

En las siguientes figuras 4 y 5 se observan los *heatmaps* para el conjunto de variantes de ClinVar, donde se representa en color rojo la clasificación patógena y en azul la benigna o neutra; en cada fila se muestran las variantes individuales y en columnas cada uno de los predictores analizados, siendo una de ellas la clasificación real con la que comparamos (*classification*). A través de este tipo de gráficos podemos ver también de forma rápida y visual los resultados analizados en la tabla anterior, esta vez analizados de forma separada según las predicciones patógena y benigna. En el primer gráfico se puede ver la gran cantidad de variantes detectadas como negativas por PrimateAI, pero que en realidad resultan ser falsos negativos, de ahí su bajo valor de sensibilidad; en el gráfico para las clasificaciones benignas se observa sin embargo la tendencia a unos valores más bajos de variantes incorrectamente detectadas como patógenas, es decir una cantidad de falsos positivos más alta.

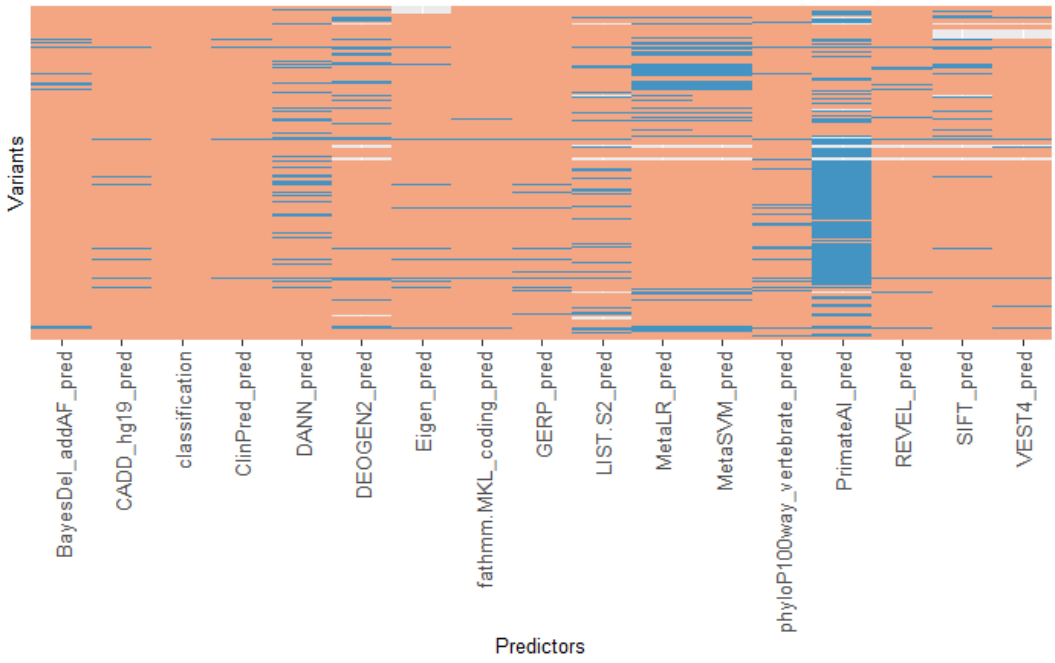


Figura 4. Heatmap que representa las predicciones de los diferentes métodos para todas las variantes patógenas del conjunto de datos de ClinVar.

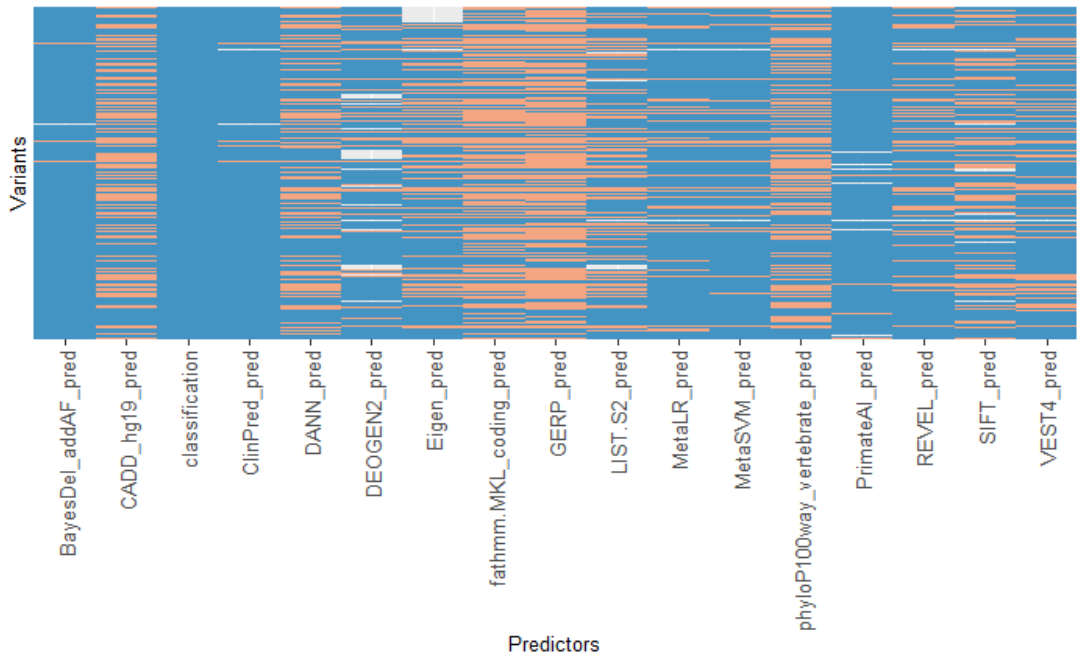


Figura 5. Heatmap que representa las predicciones de los diferentes métodos para todas las variantes benignas del conjunto de datos de ClinVar.

Para el conjunto de variantes somáticas se construyen sendos *heatmaps*, mostrados en las Figuras 6 y 7, donde podemos observar también los resultados descritos anteriormente en la tabla. Destaca especialmente como hemos comentado el predictor PrimateAI y la alta cantidad de *False Negative* que presenta, además de MetaSVM y MetaLR, que cuentan con una cantidad muy baja de *True Negatives*.

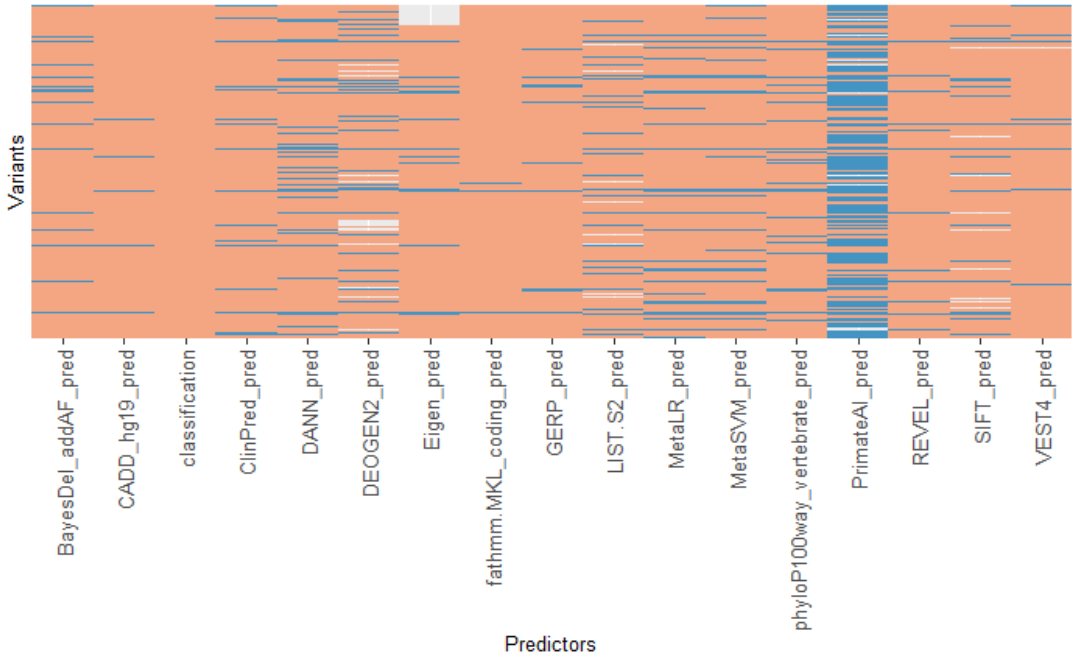


Figura 6. Heatmap que representa las predicciones de los diferentes métodos para todas las variantes patógenas del conjunto de datos de IARC/ICGC.

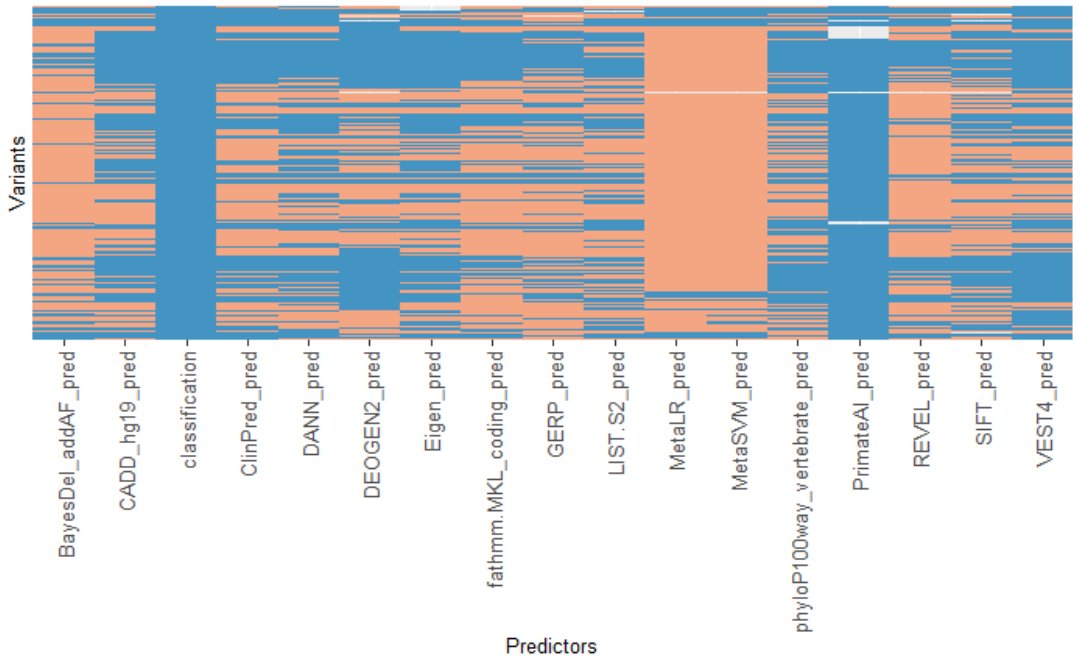


Figura 7. Heatmap que representa las predicciones de los diferentes métodos para todas las variantes benignas del conjunto de datos de IARC/ICGC.

Por otro lado, la evaluación del rendimiento de los diferentes predictores con nuestro tercer conjunto de datos, relativo a las variantes del individuo NA12878 del proyecto GIAB, nos dio también como resultado un *heatmap* con las diferentes predicciones, ya que al carecer de una clasificación real no fue posible comparar las predicciones con las métricas analizadas anteriormente. Podemos apreciar rápidamente que la mayoría de variantes posee una clasificación benigna o neutra, al aparecer en mayor medida el color azul, mientras que para ciertos predictores como GERP o phyloP, basados en el cálculo de la conservación de secuencias, aparecen en mayor medida las predicciones patógenas. Otro grupo de métodos sin embargo parece presentar un comportamiento de predicción similar, que constataremos posteriormente en las tablas de concordancia. En la siguiente Figura 8 podemos ver el *heatmap* de las predicciones para el fichero procedente de GIAB.

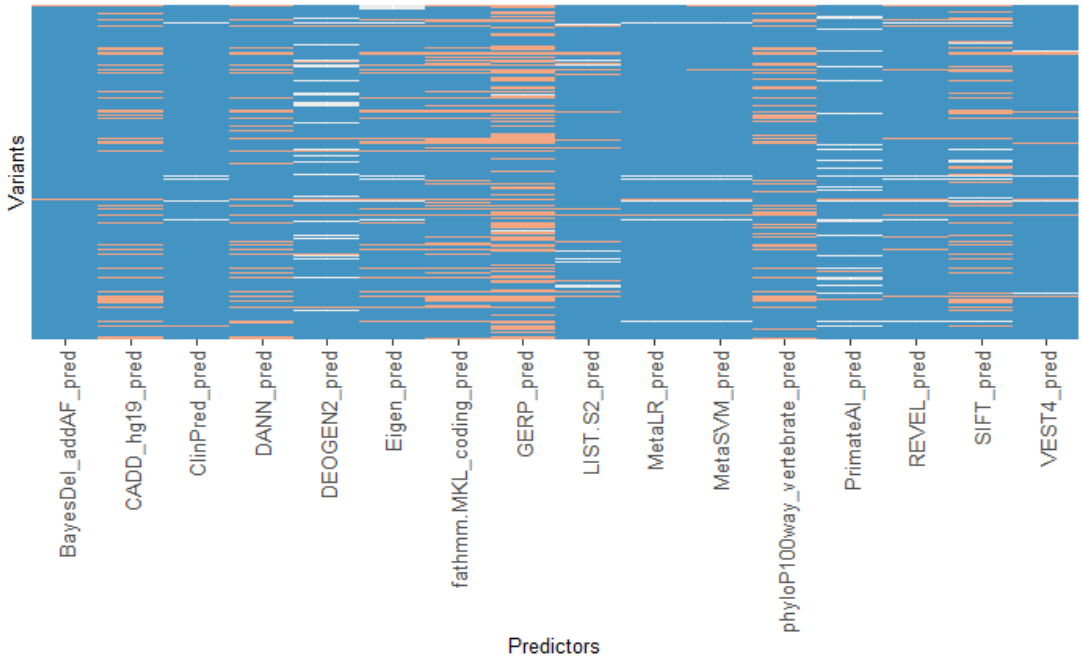


Figura 8. Heatmap que representa las predicciones de los diferentes métodos para todas las variantes del conjunto de datos del individuo NA12878 de GIAB.

Finalmente, como hemos comentado se obtienen las concordancias de los diferentes predictores en porcentajes de variantes que presentan la misma clasificación, tanto comparándolos dos a dos como para todos al mismo tiempo. Mediante este análisis se pretende conocer un poco más acerca de los grupos de predictores que pueden tener comportamientos más similares entre ellos para su posterior utilización en la práctica clínica, ya que cuanto mayor sea el porcentaje de similitud de las clasificaciones de patogenicidad predichas más sencillo será el proceso final de asignar la posible causa de una enfermedad determinada.

En primer lugar, para el conjunto de variantes de ClinVar se muestran en la Tabla 6 los diez pares de predictores con mejores porcentajes de concordancia para las variantes patógenas y benignas. Cuando se calcula el valor de similitud entre todos los predictores se obtienen valores del 19’66 % para las variantes consideradas patógenas y del 15’77 % para las benignas, esto es, el porcentaje de variantes genéticas cuyas predicciones son exactamente las mismas para todos los métodos. Este valor es relativamente bajo debido a que contamos con un gran número de métodos de predicción a analizar, con la disparidad de algoritmos y herramientas que cada uno conlleva (predicción funcional, métricas de conservación o frecuencias poblacionales), por lo que es lógico este bajo número. Si nos fijamos en las tablas se observa que aparecen muy arriba el par MetaSVM/MetaLR, ya que como se ha explicado fueron desarrollados por el mismo equipo de investigadores y se usó el mismo conjunto de datos de entrenamiento para ambos, variando únicamente el algoritmo de un *Support Vector Machine* a una *Logistic Regression*. Para las variantes benignas además aparecen con valores de similitud altos respecto a otros predictores, aunque para el grupo de las patógenas aparecen en mayor medida métodos como GERP, REVEL o fathmmMKL.

Tabla 6. Pares de predictores con mejor porcentaje de concordancia para las variantes patógenas y benignas del conjunto de datos de ClinVar.

Variantes patogenas		Variantes benignas	
Pares de predictores	**Concordancia (%)**	**Pares de predictores**	**Concordancia (%)**
MetaSVM_pred / MetaLR_pred	97.42520	**BayesDel_addAF_pred / ClinPred_pred**	97.29258
fathmm.MKL_coding_pred / CADD_hg19_pred	96.64551	**MetaSVM_pred / MetaLR_pred**	95.45705
fathmm.MKL_coding_pred / GERP_pred	95.77516	**PrimateAI_pred / ClinPred_pred**	92.82305
CADD_hg19_pred / GERP_pred	95.64823	**PrimateAI_pred / BayesDel_addAF_pred**	91.93282
fathmm.MKL_coding_pred / VEST4_pred	94.07072	**MetaSVM_pred / REVEL_pred**	89.69347
VEST4_pred / REVEL_pred	93.88939	**MetaSVM_pred / BayesDel_addAF_pred**	89.07856
fathmm.MKL_coding_pred / REVEL_pred	93.56301	**MetaSVM_pred / ClinPred_pred**	88.61050
REVEL_pred / CADD_hg19_pred	93.41795	**MetaLR_pred / REVEL_pred**	87.75697
GERP_pred / phyloP100way_vertebrate_pred	93.25476	**MetaLR_pred / BayesDel_addAF_pred**	87.39905
BayesDel_addAF / REVEL	92.96464	**MetaLR_pred / ClinPred_pred**	86.52717

En cuanto al nivel de concordancia para el conjunto de variantes somáticas, mostrado en la comparación por pares en la siguientes Tabla 7, se observa que cambia ligeramente respecto al caso anterior. Esta vez los niveles de similitud teniendo en cuenta toda la lista de métodos son de 13’77 y 0’51 % para las variantes patógenas y benignas, respectivamente, por lo que se aprecia claramente una disminución considerable en la concordancia de predicciones, especialmente para el grupo de variantes neutras. Para el conjunto de variantes patógenas los pares de predictores con mayor similitud no cambian demasiado respecto a su homólogo en el conjunto de datos de ClinVar, con métodos como fathmmMKL, CADD o GERP, mientras que para las variantes benignas aparecen métodos nuevos como DANN o Eigen; en ambos queda patente la alta concordancia entre MetaSVM y MetaLR, que no se tendrá en cuenta debido a su propia similitud interna respecto a los datos de entrenamiento para construir sus modelos.

Tabla 7. Pares de predictores con mejor porcentaje de concordancia para las variantes patógenas y benignas del conjunto de datos de IARC/ICGC.

Variantes patogenas		Variantes benignas	
Pares de predictores	**Concordancia (%)**	**Pares de predictores**	**Concordancia (%)**
MetaSVM_pred / MetaLR_pred	97.08738	**MetaSVM_pred / MetaLR_pred**	98.13559
fathmm.MKL_coding_pred / CADD_hg19_pred	96.73433	**CADD_hg19_pred / Eigen_pred**	87.11864
ClinPred_pred / CADD_hg19_pred	96.38129	**ClinPred_pred / CADD_hg19_pred**	84.74576
ClinPred_pred / fathmm.MKL_coding_pred	95.05737	**CADD_hg19_pred / DANN_pred**	83.72881
CADD_hg19_pred / Eigen_pred	94.52780	**DANN_pred / Eigen_pred**	82.71186
fathmm.MKL_coding_pred / GERP_pred	94.43954	**MetaSVM_pred / BayesDel_addAF_pred**	82.20339
fathmm.MKL_coding_pred / Eigen_pred	93.73345	**VEST4_pred / Eigen_pred**	80.84746
BayesDel_addAF_pred / ClinPred_pred	93.29214	**MetaLR_pred / BayesDel_addAF_pred**	80.33898
ClinPred_pred / Eigen_pred	92.23301	**ClinPred_pred / fathmm.MKL_coding_pred**	80.00000
BayesDel_addAF_pred / REVEL_pred	91.96823	**ClinPred_pred / DANN_pred**	79.66102

Finalmente, el análisis de concordancia para el conjunto de variantes del individuo de GIAB reporta unos resultados más acordes a lo esperado y a la realidad en el contexto de las variantes genómicas y el cálculo de su potencial patogenicidad. El valor de similitud entre todos los métodos predictores es de un 30’67 %, el más alto de los tres conjuntos de datos, y un rápido vistazo a la Tabla 8 con los pares con mayor porcentaje de concordancia nos permite ver que el análisis se ve acaparado por predictores como MetaSVM, MetaLR, REVEL, ClinPred, BayesDel y VEST4, casualmente los más novedosos y los que ocupaban los primeros puestos en las métricas de predicción de los datos de ClinVar, con valores de media más altos. Esto nos hace pensar que posiblemente para el caso clínico típico de interpretación de un exoma o panel de genes la similitud entre predictores permita una priorización de variantes más sencilla.

Tabla 8. Pares de predictores con mejor porcentaje de concordancia para las variantes del conjunto de datos de GIAB.

Pares de predictores	Concordancia (%)
MetaSVM_pred / MetaLR_pred	99.63257
MetaSVM_pred / ClinPred_pred	97.81474
MetaSVM_pred / REVEL_pred	97.67937
MetaLR_pred / REVEL_pred	97.50532
BayesDel_addAF_pred / ClinPred_pred	96.71243
MetaSVM_pred / BayesDel_addAF_pred	96.52872
MetaLR_pred / BayesDel_addAF_pred	96.35467
ClinPred_pred / REVEL_pred	96.19029
BayesDel_addAF_pred / REVEL_pred	95.50377
BayesDel_addAF_pred / VEST4_pred	94.33378

Como colofón final para este primer bloque de análisis del trabajo, dedicado a la evaluación de predictores de patogenicidad *in silico* mediante el cálculo de determinadas métricas de clasificación y el análisis de concordancia, desarrollaremos una discusión para resumir los diferentes apartados que hemos ido llevando a cabo en el proceso e intentar analizar el significado de los resultados obtenidos. Se tratará de llegar a unas conclusiones acerca de los mejores predictores de patogenicidad que pueden usarse a día de hoy en la práctica clínica, cuáles son los métodos y algoritmos que mejor funcionan en este tipo de problemas de predicción computacional o qué importancia pueden tener dentro de la etapa de priorización e interpretación de variantes en diferentes proyectos de secuenciación masiva.

Para comenzar, al analizar las dos tablas con las diferentes métricas de evaluación podemos llegar a varias conclusiones interesantes sobre los predictores, patrones y comportamientos que se dividen según el carácter patógeno o benigno de las variantes o su procedencia en cuanto al tipo de experimento. Para el conjunto de datos de ClinVar, caracterizadas por ser mayormente de carácter germinal y con un considerable nivel de fiabilidad en cuanto a las evidencias, aparecen con mejor valor de AUC los predictores ClinPred, BayesDel, REVEL, MetaSVM, MetaLR y VEST4; por otra parte, si analizamos solamente la sensibilidad, donde destacan aquellos métodos con una mayor capacidad para detectar *True* Positives o variantes realmente patógenas, aparecen VEST4 y fathmmMKL, mientras que para la especificidad, o capacidad de detectar variantes benignas, los mejores predictores son ClinPred y PrimateAI. Todos los mencionados, a excepción de VEST4 y

PrimateAI, son predictores que siguen un método ensemble, en el que además de diferentes variables utilizan como *features* las predicciones de modelos de patogenicidad existentes para entrenar su propio algoritmo y crear una clasificación de patogenicidad, optimizando en gran medida el resultado. En este sentido queda patente como el *ensemble learning* es el método con el que se obtienen mejores predicciones, por lo que su uso debería extenderse en el futuro para resolver este tipo de problemas. En cuanto a los diferentes algoritmos que utiliza cada predictor se observa como el método predominante es el *Random Forest*, usado por ClinPred, REVEL y VEST4, seguido por las máquinas de vectores soporte (SVM), por lo que parece que este algoritmo de *machine learning* es capaz de detectar de forma más precisa los patrones existentes en las variables explicativas del modelo, siendo la mayoría de ellas clasificaciones de otros predictores, características fisicoquímicas o frecuencias poblacionales. Por otro lado, uno de los conjuntos de algoritmos más en boga en los últimos años dentro del campo, como son las redes neuronales, no quedan bien representadas dentro del mundo de los predictores de patogenicidad, a excepción de PrimateAI, lo que puede deberse a la alta complejidad en el cálculo o la alta cantidad de datos que se necesitan para entrenar este tipo de modelos, algo muy difícil de conseguir en el campo de la genómica clínica.

Cuando estudiamos el valor de las métricas para el conjunto de variantes somáticas vemos que surgen patrones ligeramente diferentes. Los métodos con mejores valores de área bajo la curva son VEST4, Eigen, CADD y ClinPred; para la detección de variantes patógenas aparecen ClinPred y fathmmMKL, mientras que para la especificidad destaca de nuevo por encima de todos PrimateAI. Un aspecto a destacar respecto a este conjunto de variantes son los pobres resultados que presentan tanto MetaSVM como MetaLR, con valores de AUC por debajo del umbral de 0’50 y especificidades del orden de 0’11 y 0’10, respectivamente; esta anomalía no puede explicarse de otra forma que no sea un error en la medida de los datos o en la obtención de las predicciones, sin descartar que el propio repositorio dbNSFP, que tan buenos resultados nos ha dado, pueda contar con predicciones erróneas que no se correspondan con las de las herramientas. Es prácticamente imposible que la predicción de variantes benignas sea tan pobre para estos dos métodos, especialmente teniendo en cuenta que para las variantes germinales de ClinVar sí contaban con predicciones lógicas.

A tenor de los datos presentados podemos decir que existen ciertas conclusiones o patrones que pueden extraerse del análisis en base a las métricas de clasificación. En primer lugar, para un experimento genérico de secuenciación masiva donde la interpretación de variantes clínicas sea importante, un conjunto de predictores óptimo debería estar formado por ClinPred, BayesDel, REVEL, VEST4, fathmmMKL y PrimateAI, tratando de estudiar cada caso específico para establecer la clasificación más adecuada. Los primeros cuatro métodos son los que en general, para casi todas las condiciones estudiadas, ofrecen mejores resultados, por lo que siempre deberían considerarse como opción preferente en la predicción de variantes; por otro lado, fathmmMKL y PrimateAI son dos métodos que complementan muy bien al resto dependiendo del enfoque en el que estemos trabajando, ya

que sus clasificaciones están muy sesgadas hacia un tipo de variante u otro. En este caso, fathmmMKL tiene una muy buena capacidad de detectar correctamente el carácter patógeno de las variantes estudiadas, mientras que PrimateAI sigue un comportamiento similar respecto a las variantes benignas, por lo que pueden servir para validar o confirmar una clasificación determinada previamente por los otros predictores. Teniendo en cuenta que la predicción para el grupo de variantes contrario es bastante deficiente (detección de variantes patógenas para PrimateAI y benignas para fathmmMKL), las razones de sus comportamientos tan inusuales pueden ser muy variadas. Por un lado, la inclusión en el modelo de fathmmMKL de propiedades epigenéticas y el contexto genómico de las variantes, algo exclusivo de este predictor, puede hacer que tenga una mayor capacidad de detección de la patogenicidad de una región genómica, debido a la importancia que tiene este contexto epigenético en la expresión y función de la mayoría de genes; por otro lado, la construcción del modelo de PrimateAI como herramienta de comparación de secuencias de primates hace que sea capaz de encontrar un mayor número de variantes benignas de lo normal, demostrando el valor que presenta la evolución comparativa en especies relativamente cercanas, ya que el hecho de que un polimorfismo genético sea neutro en este tipo de especies aumenta las probabilidades de que lo sea en seres humanos, por la alta similitud de las secuencias proteicas.

El análisis de los *heatmaps* o mapas de calor nos complementa y potencia la visión descrita anteriormente para el estudio de las métricas estadísticas, ya que se aprecia claramente el comportamiento de cada uno de los predictores en función del resto. Un aspecto a tener en cuenta respecto a este tipo de gráficos es que nos permiten visualizar rápidamente la capacidad de clasificación y concordancia para entornos y variantes distintos, como es el caso de nuestros tres conjuntos de datos. Las predicciones sufren considerablemente al pasar al conjunto de variantes procedente de los repositorios IARC e ICGC, cuyo carácter somático las hace muy difíciles de detectar, clasificar y caracterizar, por lo que en muchas ocasiones las predicciones de patogenicidad no resultan del todo fiables. Es necesario por tanto poner énfasis en este tipo de variantes mediante el desarrollo y el estudio de nuevos algoritmos que tengan en cuenta variables intrínsecas de este tipo de variantes, tan esenciales actualmente en el diagnóstico y mejora del pronóstico de numerosos tipos de tumores. Por otro lado, el patrón de predicciones que observamos en el tercer conjunto de datos, relativo a las variantes procedentes del proyecto de *Genome in a Bottle* que representan el caso de estudio estándar, es el que podríamos encontrarnos con total seguridad en un típico caso de secuenciación con fines clínicos. Como se puede apreciar, la proporción de variantes benignas o neutras supera en gran medida la de variantes patógenas, algo lógico si pensamos en la propia teoría de la evolución; sin embargo, la importancia de este tipo de experimentos reside en la capacidad de encontrar aquella variante que con mayor probabilidad es la causa de la enfermedad que se estudia, por lo que es en este punto donde deben centrarse todos los predictores desarrollados, en contar con una alta fiabilidad para detectar *True Positives* y por tanto presentar un muy buen valor de *Sensitivity*.

Finalmente, el análisis de concordancia mediante el análisis de similitud refuerza la idea de utilizar un conjunto de predictores como método óptimo para encontrar la clasificación final de una variante, especialmente de los métodos mencionados anteriormente. A excepción de MetaSVM y MetaLR, que con su alta similitud deben despreciarse al proceder del mismo grupo de desarrollo, la mayoría de predictores mencionados como óptimos aparecen en con un alto valor de concordancia, lo que facilita en gran medida el establecimiento de una predicción conjunta.

3.2 Evaluación plataformas de priorización de variantes genéticas

En esta última parte del proyecto se llevaron a cabo los análisis correspondientes en las diferentes plataformas de priorización de variantes, además de recoger y presentar los principales resultados que se muestran para explicar con exactitud qué ventajas e inconvenientes poseen cada uno de ellos. En la medida de lo posible se intentó analizar e interpretar los conjuntos de variantes que hemos escogido en la primera parte del proyecto, como son las variantes del repositorio ClinVar, el conjunto de variantes somáticas de TP53 e ICGC y los datos de la secuenciación del individuo NA12878 del proyecto *Genome in a Bottle*, aunque como veremos en alguno de los casos no ha sido posible introducirlos como *input* o ha habido que realizar determinados preprocesamientos para que encajaran en los parámetros de cada una.

- VarCards

La primera herramienta de priorización descrita anteriormente, VarCards, nos da como resultado un fichero en formato TSV con las diferentes anotaciones y campos asociados a cada variante que hayamos introducido, por lo que es muy sencillo abrir el fichero en local para visualizarlo rápidamente. Llega a ofrecer hasta 131 anotaciones distintas, especialmente predictores de patogenicidad *in silico* y frecuencias alélicas en diferentes subpoblaciones y proyectos, como son los 1000 genomas o la *gnomAD initiative*, sin dejar de presentar la información más básica como coordenadas genómicas, cambio de aminoácido y situación dentro de un gen en cuestión. La información dentro de los campos se muestra muy limpia, clara y fácil de entender, sin tener que averiguar el significado de diferentes términos o símbolos dependiendo de cada predictor, ya que cada uno utiliza un sistema de anotación y un umbral de patogenicidad diferente. Junto con este documento se adjuntarán en el Anexo las tres tablas de resultados que hemos obtenido con VarCards para dos de los conjuntos de datos analizados, como son las variantes de ClinVar y las pertenecientes al individuo de *Genome in a Bottle*; el conjunto de variantes somáticas no se analizó por la simple razón de que se obtendrían unos resultados que no aportarían nada a la discusión, pues es un *dataset* con unas propiedades y características similares al conjunto de datos de ClinVar, con información real sobre patogenicidad con la que evaluar los predictores del bloque anterior, y en este caso pretendemos analizar las herramientas desde otro tipo de enfoque.

A continuación se muestra una imagen de la pantalla inicial de la pestaña '*Annotate*' dentro de la plataforma web de VarCards, véase (93) y Figura 9, donde aparecen en primer lugar las opciones para introducir nuestras variantes

en formato VCF y la dirección de email donde queremos que nos lleguen los resultados, mientras que más abajo podemos seleccionar los diferentes campos de anotación que presentará la tabla de resultados. Se observa también que es posible filtrar variantes según las frecuencias poblacionales estableciendo umbrales para determinar si una variante es extrema o muy poco común en la población.

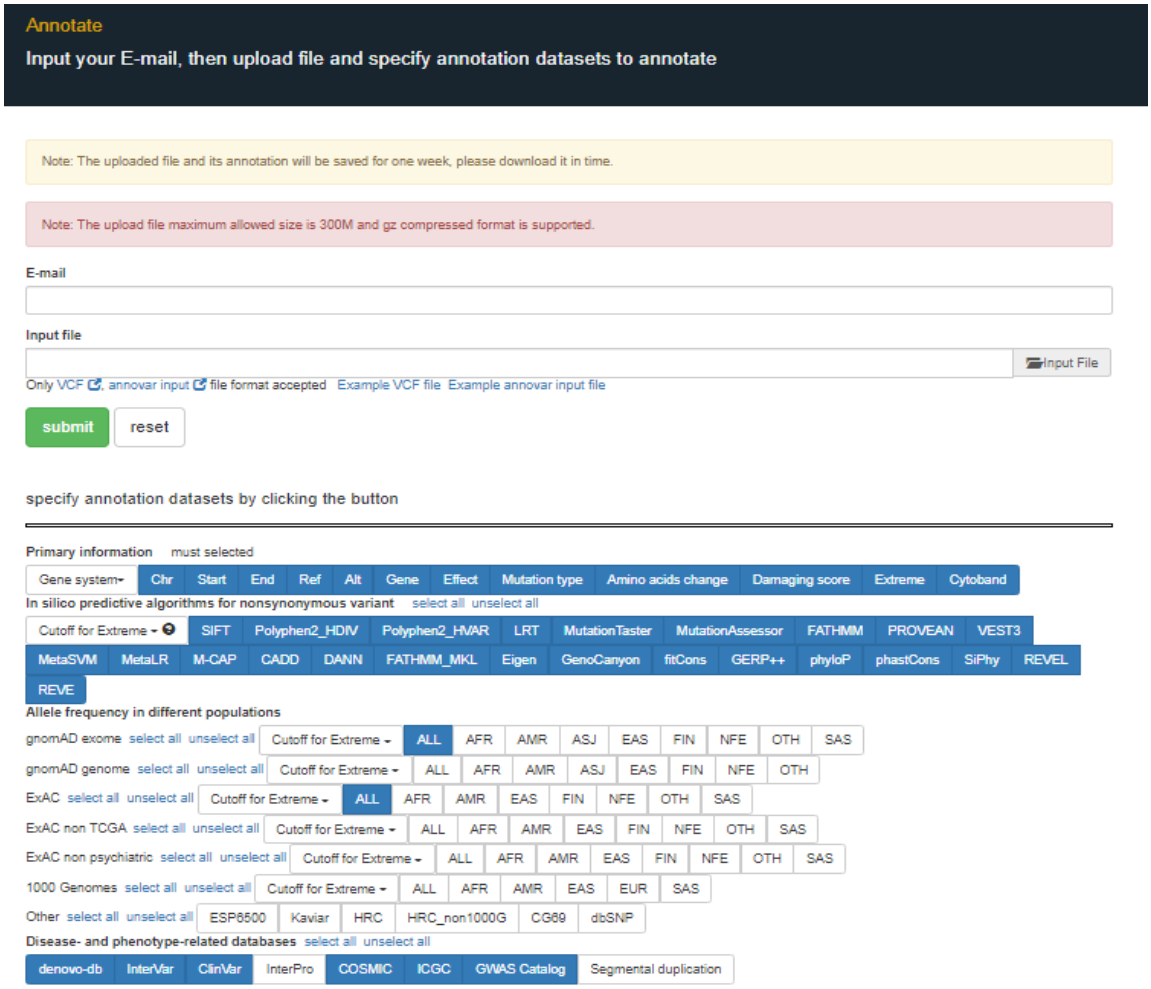


Figura 9. Captura de pantalla representando la disposición de las distintas opciones y parámetros de la pestaña Annotate de VarCards.

- GenIO

La plataforma de priorización de variantes GenIO (94), cuyo enfoque se basa en incluir información clínica en base a posibles enfermedades causantes del fenotipo o a síntomas observados, posee una apariencia en la primera pantalla como la que se observa en la siguiente Figura 10. En ella podemos introducir nuestro conjunto de variantes en formato VCF, la dirección de email donde queremos que nos lleguen los resultados del análisis y los diferentes parámetros y características del estudio que debemos introducir, como son los posibles síntomas, enfermedades (con la capacidad de autocompletado) o descubrimientos complementarios a tener en cuenta, además de algunas opciones avanzadas como la posibilidad de establecer el umbral de rareza de nuestras variantes o la lista de genes que pretendemos estudiar.

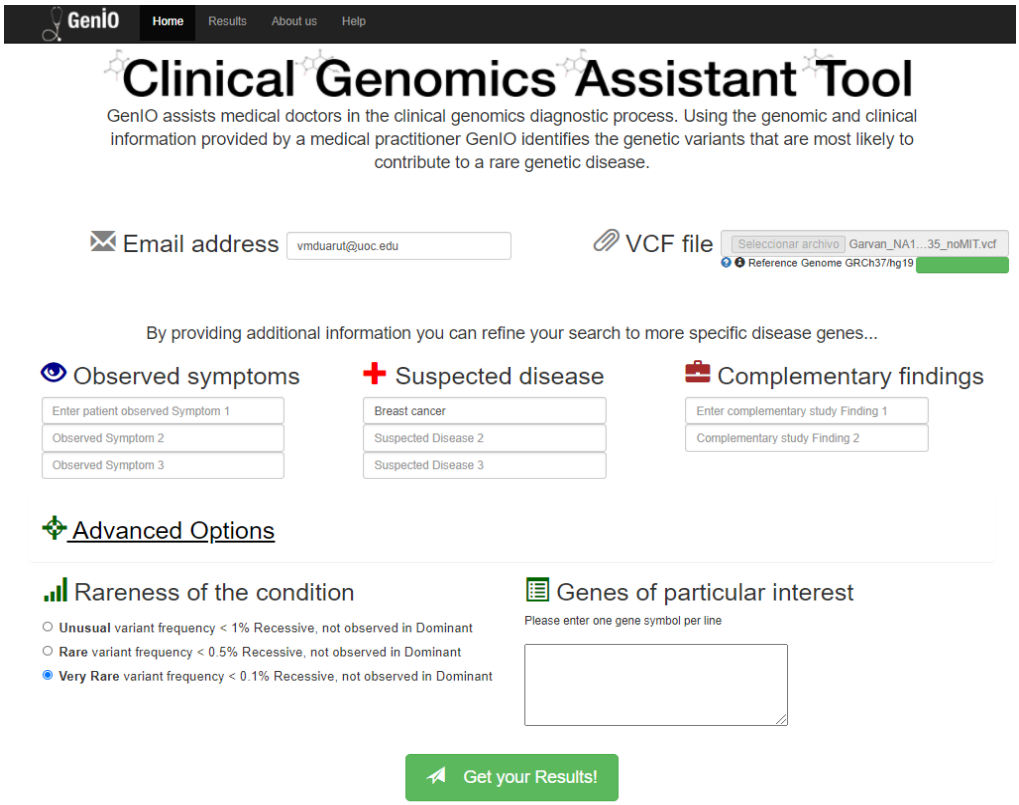


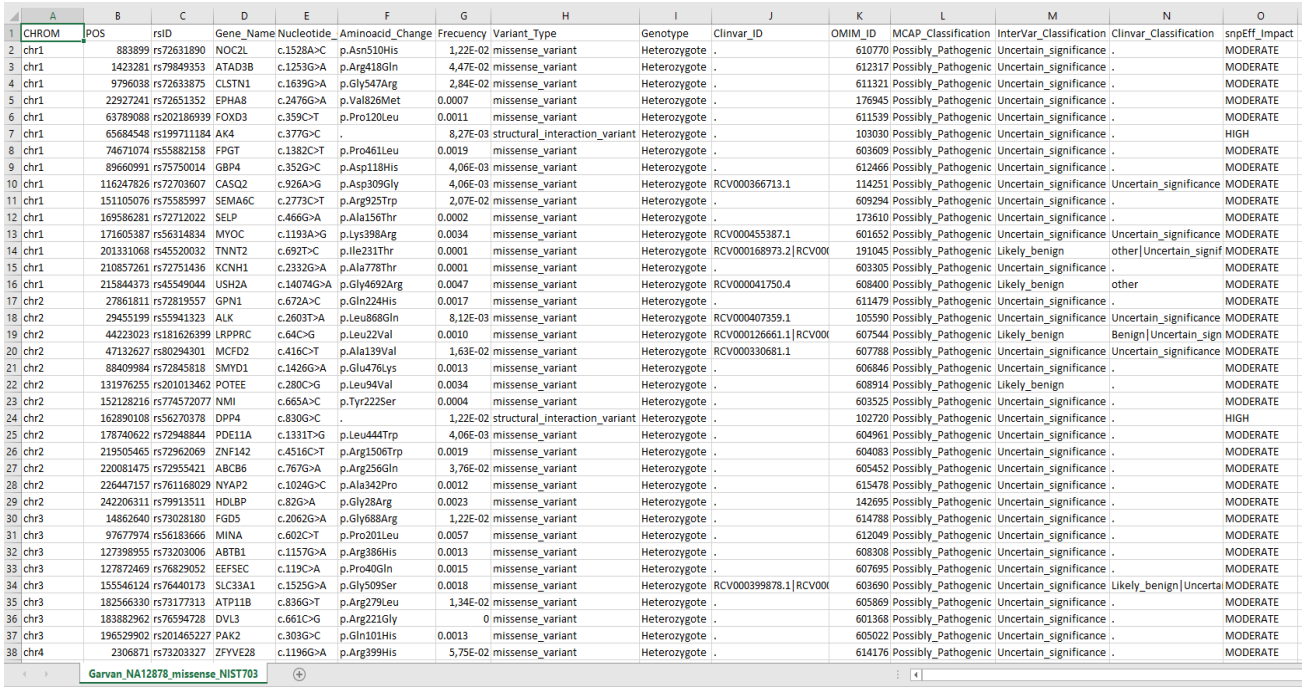
Figura 10. Captura de pantalla representando la página inicial de GenIO, con las diferentes opciones para introducir el fichero VCF, la dirección de email y los diferentes parámetros para priorizar.

Hay que resaltar que para esta plataforma solamente se han analizado las variantes pertenecientes al individuo NA12878 del proyecto *Genome in a Bottle*, ya que era el único que cumplía con los requerimientos de GenIO en cuanto a campos del fichero VCF, la presencia de información genotípica en diferentes individuos para ajustar la lista priorizada de variantes, por lo que sólo se obtuvieron resultados para este conjunto de datos.

Tras el análisis realizado estableciendo el cáncer de mama, o *breast cancer*, como posible enfermedad se obtienen en la pantalla de resultados varios apartados, entre los que podemos destacar dos de ellos: un primer bloque con información de variantes descubiertas según los modelos de recesividad y dominancia, y un segundo bloque con dos subapartados, uno para descargar los resultados en local y obtener la tabla con las anotaciones completas y otro para visualizar la lista priorizada de variantes potencialmente patógenas, con información sobre las diferentes predicciones. El fichero VCF con las anotaciones completas para cada variante es bastante interesante, ya que ofrece hasta 183 campos distintos con información muy diversa, desde datos de frecuencias poblacionales hasta cálculos de las distintas métricas acordadas por la ACMG para la interpretación de variantes (95). Junto a este documento se adjuntarán los resultados obtenidos incluyendo el VCF completo con las anotaciones.

En cuanto a la lista priorizada, vemos en la siguiente Figura 11 los diferentes campos que nos ofrece para valorar la patogenicidad de ciertas variantes,

filtradas por la herramienta en base a varios factores como la frecuencia alélica en poblaciones, la presencia del gen en la base de datos OMIM, la interpretación de cada variante según el predictor M-CAP e InterVar o el potencial impacto que causa dicha variante en la proteína.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	CHROM	POS	rsID	Gene_Name	Nucleotide_	Aminoacid_Change	Frecuency	Variant_Type	Genotype	Clinvar_ID	OMIM_ID	MCAP_Classification	InterVar_Classification	Clinvar_Classification	snpEff_Impact
2	chr1	883899	rs72631890	NOC2L	c.1528A>C	p.Asn510His	1,22E-02	missense_variant	Heterozygote	.	610770	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
3	chr1	1423281	rs79849353	ATAD3B	c.1253G>A	p.Arg418Gln	4,47E-02	missense_variant	Heterozygote	.	612317	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
4	chr1	9796038	rs72633875	CLSTN1	c.1639G>A	p.Gly547Arg	2,84E-02	missense_variant	Heterozygote	.	611321	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
5	chr1	22927241	rs72651352	EPHA8	c.2476G>A	p.Val826Met	0.0007	missense_variant	Heterozygote	.	176945	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
6	chr1	63789088	rs202186939	FOXD3	c.359C>T	p.Pro120Leu	0.0011	missense_variant	Heterozygote	.	611539	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
7	chr1	65684548	rs199711184	AK4	c.377G>C	.	8,27E-03	structural_interaction_variant	Heterozygote	.	103030	Possibly_Pathogenic	Uncertain_significance	.	HIGH
8	chr1	74671074	rs55882158	FPGT	c.1382C>T	p.Pro461Leu	0.0019	missense_variant	Heterozygote	.	603609	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
9	chr1	89660991	rs75750014	GBP4	c.352G>C	p.Asp118His	4,06E-03	missense_variant	Heterozygote	.	612466	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
10	chr1	116247826	rs72703607	CASQ2	c.926A>G	p.Asp309Gly	4,06E-03	missense_variant	Heterozygote	RCV000366713.1	114251	Possibly_Pathogenic	Uncertain_significance	Uncertain_significance	MODERATE
11	chr1	151105076	rs75585997	SEMA6C	c.2773C>T	p.Arg925Trp	2,07E-02	missense_variant	Heterozygote	.	609294	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
12	chr1	169586281	rs72712022	SELP	c.466G>A	p.Ala156Thr	0.0002	missense_variant	Heterozygote	.	173610	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
13	chr1	171605387	rs56314834	MYOC	c.1193A>G	p.Lys398Arg	0.0034	missense_variant	Heterozygote	RCV000455387.1	601652	Possibly_Pathogenic	Uncertain_significance	Uncertain_significance	MODERATE
14	chr1	201331068	rs45520032	TNNT2	c.692T>C	p.Ile231Thr	0.0001	missense_variant	Heterozygote	RCV000168973.2|RCV00(	191045	Possibly_Pathogenic	Likely_benign	other|Uncertain_signif	MODERATE
15	chr1	210857261	rs72751436	KCNH1	c.2332G>A	p.Ala778Thr	0.0001	missense_variant	Heterozygote	.	603305	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
16	chr1	215844373	rs45549044	USH2A	c.14074G>A	p.Gly4692Arg	0.0047	missense_variant	Heterozygote	RCV000041750.4	608400	Possibly_Pathogenic	Likely_benign	other	MODERATE
17	chr2	27861811	rs72819557	GPN1	c.672A>C	p.Gln224His	0.0017	missense_variant	Heterozygote	.	611479	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
18	chr2	29455199	rs55941323	ALK	c.2603T>A	p.Leu868Gln	8,12E-03	missense_variant	Heterozygote	RCV000407359.1	105590	Possibly_Pathogenic	Uncertain_significance	Uncertain_significance	MODERATE
19	chr2	44223023	rs181626399	LRPPRC	c.64C>G	p.Leu22Val	0.0010	missense_variant	Heterozygote	RCV000126661.1|RCV00(	607544	Possibly_Pathogenic	Likely_benign	Benign|Uncertain_sign	MODERATE
20	chr2	47132627	rs80294301	MCFD2	c.416C>T	p.Ala139Val	1,63E-02	missense_variant	Heterozygote	RCV000330681.1	607788	Possibly_Pathogenic	Uncertain_significance	Uncertain_significance	MODERATE
21	chr2	88409984	rs72845818	SMYD1	c.1426G>A	p.Glu476Lys	0.0013	missense_variant	Heterozygote	.	606846	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
22	chr2	131976255	rs201013462	POTEE	c.280C>G	p.Leu94Val	0.0034	missense_variant	Heterozygote	.	608914	Possibly_Pathogenic	Likely_benign	.	MODERATE
23	chr2	152128216	rs774572077	NMI	c.665A>C	p.Tyr222Ser	0.0004	missense_variant	Heterozygote	.	603525	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
24	chr2	162890108	rs56270378	DPP4	c.830G>C	.	1,22E-02	structural_interaction_variant	Heterozygote	.	102720	Possibly_Pathogenic	Uncertain_significance	.	HIGH
25	chr2	178740622	rs72948844	PDE11A	c.1331T>G	p.Leu444Trp	4,06E-03	missense_variant	Heterozygote	.	604961	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
26	chr2	219505465	rs72962069	ZNF142	c.4516C>T	p.Arg1506Trp	0.0019	missense_variant	Heterozygote	.	604083	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
27	chr2	220081475	rs72955421	ABCB6	c.767G>A	p.Arg256Gln	3,76E-02	missense_variant	Heterozygote	.	605452	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
28	chr2	226447157	rs761168029	NYAP2	c.1024G>C	p.Ala342Pro	0.0012	missense_variant	Heterozygote	.	615478	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
29	chr2	242206311	rs79913511	HDLBP	c.82G>A	p.Gly28Arg	0.0023	missense_variant	Heterozygote	.	142695	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
30	chr3	14862640	rs73028180	FGD5	c.2062G>A	p.Gly688Arg	1,22E-02	missense_variant	Heterozygote	.	614788	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
31	chr3	97677974	rs56183666	MINA	c.602C>T	p.Pro201Leu	0.0057	missense_variant	Heterozygote	.	612049	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
32	chr3	127398955	rs73203006	ABTB1	c.1157G>A	p.Arg386His	0.0013	missense_variant	Heterozygote	.	608308	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
33	chr3	127872469	rs76829052	EEFSEC	c.119C>A	p.Pro40Gln	0.0015	missense_variant	Heterozygote	.	607695	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
34	chr3	155546124	rs76440173	SLC33A1	c.1525G>A	p.Gly509Ser	0.0018	missense_variant	Heterozygote	RCV000399878.1|RCV00(	603690	Possibly_Pathogenic	Uncertain_significance	Likely_benign|Uncertai	MODERATE
35	chr3	182566330	rs73177313	ATP11B	c.836G>T	p.Arg279Leu	1,34E-02	missense_variant	Heterozygote	.	605869	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
36	chr3	183882962	rs76594728	DVL3	c.661C>G	p.Arg221Gly	0	missense_variant	Heterozygote	.	601368	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
37	chr3	196529902	rs201465227	PAK2	c.303G>C	p.Gln101His	0.0013	missense_variant	Heterozygote	.	605022	Possibly_Pathogenic	Uncertain_significance	.	MODERATE
38	chr4	2306871	rs73203327	ZFYVE28	c.1196G>A	p.Arg399His	5,75E-02	missense_variant	Heterozygote	.	614176	Possibly_Pathogenic	Uncertain_significance	.	MODERATE

Garvan_NA12878_missense_NIST703

Figura 11. Muestra del fichero resultante de obtener la lista de variantes priorizada por GenIO en formato Excel, con algunos de los campos de anotaciones que presenta.

- MutationDistiller

En tercer lugar, MutationDistiller (96) es capaz de obtener una lista de variantes genéticas priorizadas según varios factores, como ya explicamos anteriormente, entre los cuales tenemos el fenotipo introducido, la región génica a estudiar, el tipo de variante, el modo de herencia o la predicción de patogenicidad basada en el algoritmo MutationTaster. Todos estos factores se pueden observar en la siguiente Figura 12, que se corresponde con la pantalla inicial donde se ajustan los diferentes parámetros para llevar a cabo el análisis de priorización, entre los cuales se puede apreciar justo en el centro el recuadro para introducir los términos ontológicos relativos al fenotipo o conjunto de fenotipos asociado con una determinada patología. Previamente ya se ha introducido en una pantalla anterior el fichero VCF con las variantes a analizar, que en este caso al igual que con GenIO hemos añadido el fichero correspondiente a las variantes de NA12878, al cumplir también con los requisitos de la plataforma (información genotípica esencialmente).

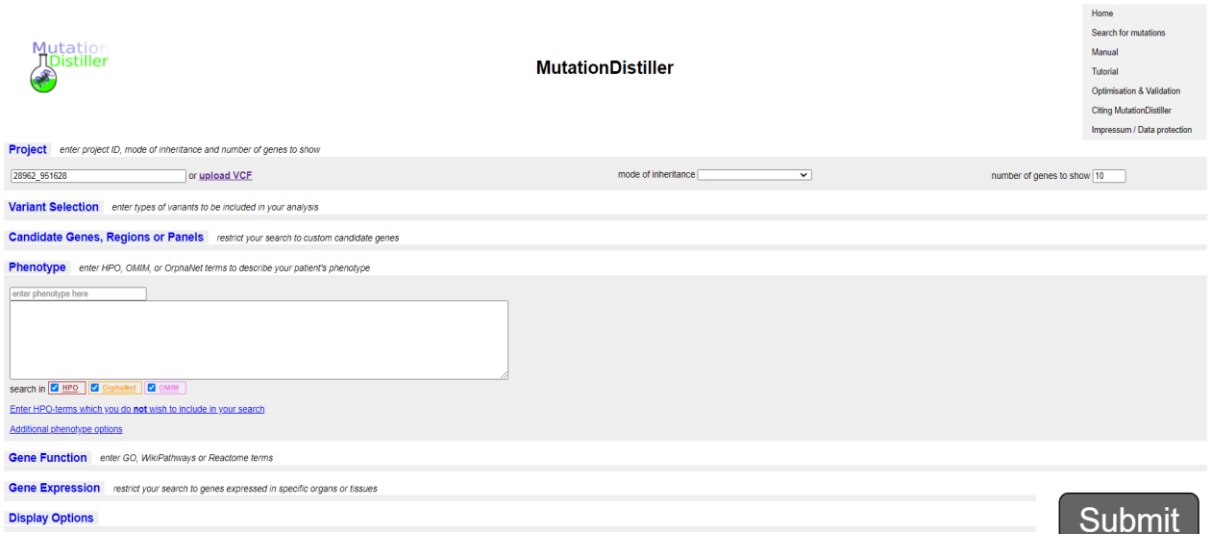
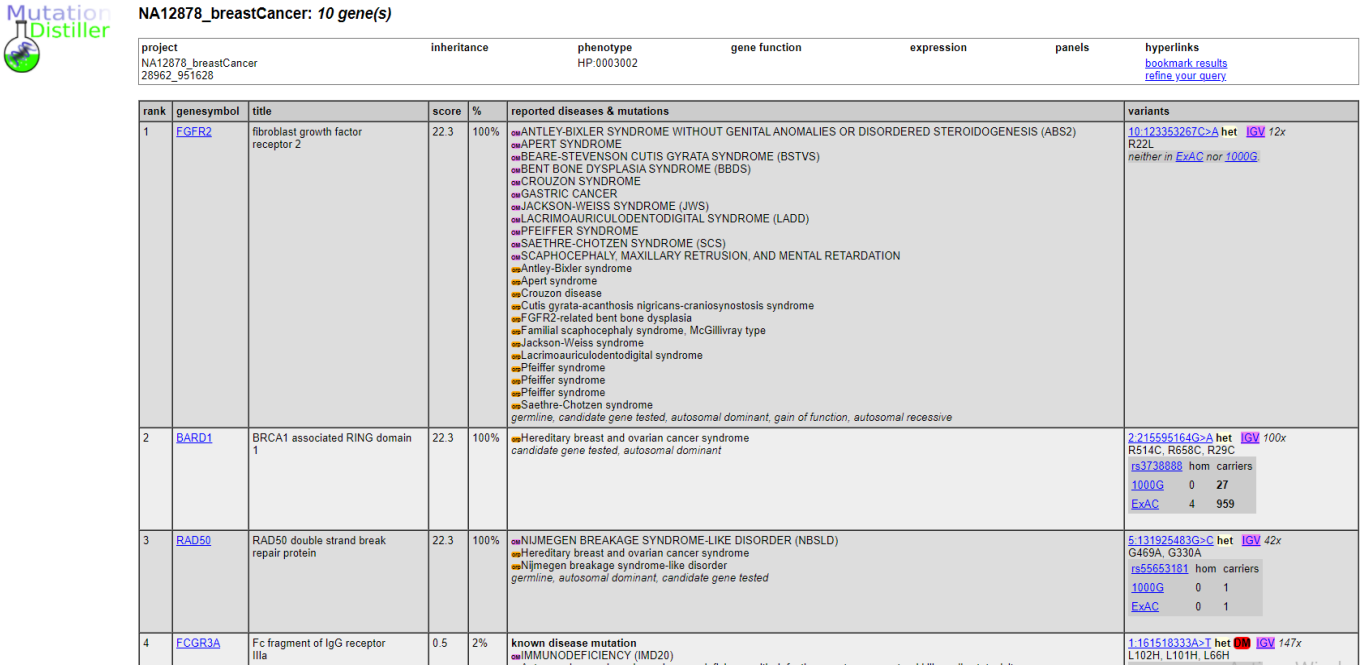


Figura 12. Captura de pantalla mostrando las diferentes opciones y parámetros de priorización de MutationDistiller.

En la siguiente Figura 13 podemos observar cómo se presentan los resultados una vez nos avisan a través de un correo electrónico, al igual que la mayoría de plataformas. La herramienta como ya comentamos nos ofrece una lista de las variantes con mayor potencial de causar la enfermedad o fenotipo introducido, que para el caso también se ha escogido el cáncer de mama (*Breast carcinoma* según HPO), con información variada sobre los fenotipos asociados al gen en cuestión, el score obtenido o datos sobre frecuencias alélicas; más abajo sin embargo, donde acaba la lista de los 10 genes con mayor puntuación (parámetro que podemos modificar en la pantalla inicial) se añade información adicional sobre la variante y el gen en el que se localizan para completar la interpretación, como datos sobre reacciones metabólicas donde tiene lugar a partir de KEGG (97) o Reactome (98,99), información sobre genes parálogos o enlaces a diferentes repositorios. Junto a esta memoria se adjuntará también el fichero resultante de descargar la lista de variantes priorizadas en formato CSV, con información básica que nos describen algunas características de la variante o el gen en cuestión.



NA12878_breastCancer: ***10 gene(s)***

project	inheritance	phenotype	gene function	expression	panels	hyperlinks
NA12878_breastCancer 28962_951628		HP:0003002				bookmark results refine your query

rank	genesymbol	title	score	%	reported diseases & mutations	variants
1	FGFR2	fibroblast growth factor receptor 2	22.3	100%	ANTLEY-BIXLER SYNDROME WITHOUT GENITAL ANOMALIES OR DISORDERED STEROIDOGENESIS (ABS2) APERT SYNDROME BEARE-STEVENSON CUTIS GYRATA SYNDROME (BSTVS) BENT BONE DYSPLASIA SYNDROME (BBDS) CROUZON SYNDROME GASTRIC CANCER JACKSON-WEISS SYNDROME (JWS) LACRIMOAURICULODENTODIGITAL SYNDROME (LADD) PFEIFFER SYNDROME SAETHRE-CHOTZEN SYNDROME (SCS) SCAPHOCEPHALY, MAXILLARY RETRUSION, AND MENTAL RETARDATION Antley-Bixler syndrome Apert syndrome Crouzon disease Cutis gyrata-acanthosis nigricans-craniosynostosis syndrome FGFR2-related bent bone dysplasia Familial scaphocephaly syndrome, McGillivray type Jackson-Weiss syndrome Lacrimoauriculodentodigital syndrome Pfeiffer syndrome Pfeiffer syndrome Pfeiffer syndrome Saethre-Chotzen syndrome *germline, candidate gene tested, autosomal dominant, gain of function, autosomal recessive*	10:123353267C>A het IGV *12x* R22L *neither in ExAC nor 1000G*
2	BARD1	BRCA1 associated RING domain 1	22.3	100%	Hereditary breast and ovarian cancer syndrome *candidate gene tested, autosomal dominant*	2:215595164G>A het IGV *100x* R514C, R658C, R29C rs3738888 hom carriers 1000G 0 27 ExAC 4 959
3	RAD50	RAD50 double strand break repair protein	22.3	100%	NIJMEGEN BREAKAGE SYNDROME-LIKE DISORDER (NBSLD) Hereditary breast and ovarian cancer syndrome Nijmegen breakage syndrome-like disorder *germline, autosomal dominant, candidate gene tested*	5:131925483G>C het IGV *42x* G469A, G330A rs55653181 hom carriers 1000G 0 1 ExAC 0 1
4	FCGR3A	Fc fragment of IgG receptor IIIa	0.5	2%	**known disease mutation** IMMUNODEFICIENCY (IMD20)	1:161518333A>T het OMI IGV *147x* L102H, L101H, L66H

Figura 13. Captura de pantalla mostrando cómo aparecen los resultados de la priorización de variantes, con el correspondiente score, enfermedades relacionadas y enlaces a diferentes bases de datos.

- OpenCRAVAT

La última herramienta de interpretación de variantes analizada en este trabajo, conocida como OpenCRAVAT, sigue un enfoque de análisis similar al de VarCards que vimos al inicio, ya que es una plataforma que mediante la selección de diversos parámetros y variables es capaz de anotar de forma exhaustiva el conjunto de variantes introducido, llegando al nivel de obtener hasta 156 campos de anotación distintos para describir y caracterizar a las variantes genómicas. Como en este caso la plataforma no establece ningún requisito de presencia de información genotípica hemos podido llevar a cabo la interpretación para los conjuntos de datos de ClinVar, tanto las variantes patógenas como las benignas, y para el individuo NA12878 de GIAB.

La pantalla de resultados se divide en varias pestañas, incluyendo información básica en forma de resumen con diferentes gráficas, una herramienta avanzada para filtrar la información de salida, la tabla completa con todas las anotaciones obtenidas para cada variante e información sobre los genes detectados en dichas variantes. Uno de los aspectos más interesante se puede observar en la siguiente Figura 14, donde se muestra una captura de los diferentes gráficos que se muestran en el *Summary* para hacernos una idea de alguno de los aspectos fundamentales en cuanto a la interpretación de variantes, como por ejemplo los porcentajes de consecuencias, una red génica de interacción o un *circos plot* con el número de variantes por cada posición genómica, todo ello de una manera visualmente atractiva para el usuario. Es posible además modificar la disposición de los gráficos en esta pantalla, desplazándolos a nuestro antojo y eliminando o añadiendo los que nos interesen. Posteriormente, la pestaña de filtro nos permite seleccionar de forma rápida qué resultados queremos que se muestren por pantalla, estableciendo

qué regiones génicas nos interesan o umbrales de patogenicidad para diferentes predictores.

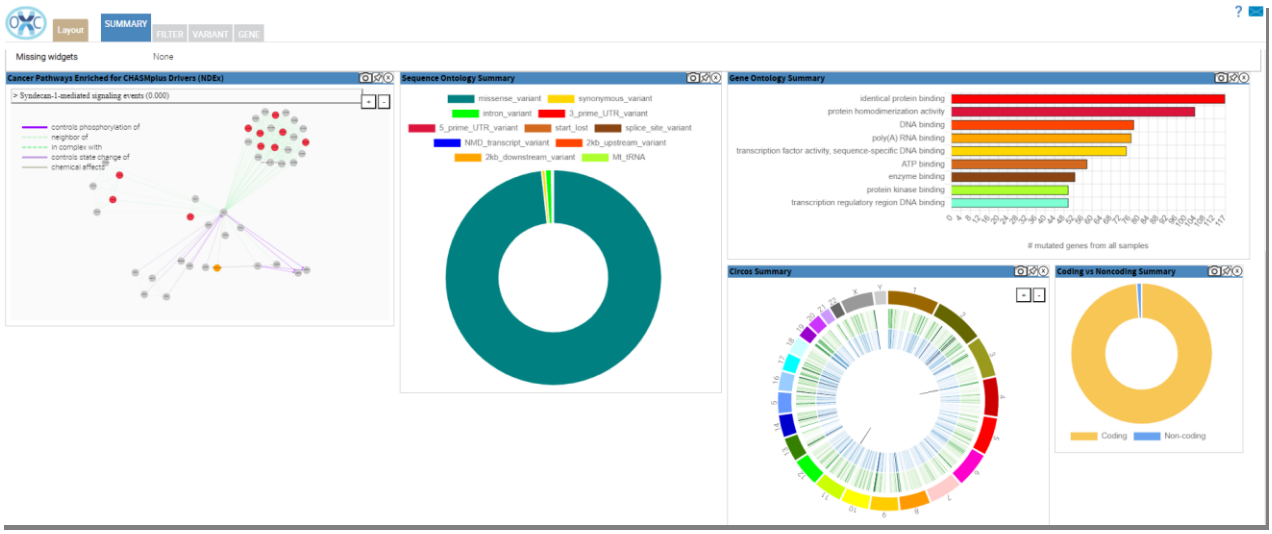


Figura 14. Captura de pantalla mostrando la pestaña Summary de OpenCRAVAT, con los diferentes gráficos que ofrece la herramienta.

Finalmente nos encontramos con la pestaña más interesante y donde aparecen los resultados completos en forma de tabla, mostrando todas y cada una de las variantes con sus anotaciones de forma clara y precisa; es posible también modificar la disposición de esta pestaña para mostrar por debajo de la tabla las anotaciones de cada variante divididas por bloques funcionales, de tal manera que podemos reunir toda la información que interesa de una variante en concreto. La cantidad de anotaciones y campos diferentes que nos ofrece esta plataforma es inmensa, desde todo tipo de predictores computacionales de patogenicidad, pasando por frecuencias alélicas en los principales repositorios globales hasta anotaciones específicas de determinadas condiciones, como el predictor de variantes tumorales CHASM (38).

Al igual que para el resto de plataformas adjuntaremos junto a este documento los resultados en forma de tablas con formato TSV, tanto para las variantes de ClinVar como para las de *Genome in a Bottle*.

3.3 Discusión de la comparación de plataformas de priorización

Como hemos podido observar existen multitud de plataformas y herramientas dedicadas al proceso de interpretación y/o priorización de variantes genéticas, cada una con diferentes características que las hacen muy complejas de clasificar y poder evaluar de forma estandarizada. Ha sido una de las complicaciones que nos hemos encontrado en el transcurso de este segundo bloque de análisis, ya que no ha sido posible comparar los resultados de las plataformas escogidas por los diferentes enfoques que sigue cada una, las posibilidades que ofrece al usuario, el método de priorización en caso de que lo tengan o la variedad de anotaciones biológicas que ofrecen. Por tanto, el valor de este bloque de análisis en nuestro trabajo reside en la evaluación subjetiva de las plataformas en base a todos los factores mencionados, intentando establecer qué métodos son los más apropiados para la interpretación de

variantes en entornos clínicos en base a nuestros conjuntos de datos; no obstante, hay que tener en cuenta que el presente análisis sólo comprende cuatro plataformas de interpretación de entre las muchas existentes en la bibliografía, debido al relativo alcance del proyecto, por lo que las conclusiones que se obtengan no deben interpretarse como reglas de oro para la elección de este tipo de herramientas sino como una base o punto de partida para futuros estudios, además de servir de guía básica sobre qué aspectos a tener en cuenta para su elección y uso.

Una vez hecha esta distinción, es preciso matizar que la evaluación o comparación se realizará por pares, es decir, dividiremos las cuatro plataformas de priorización en dos enfoques diferentes y haremos la comparación objetiva dentro de estos dos grupos, ya que como decíamos antes la alta diversidad de métodos hace que sea difícil una evaluación estándar. De esta forma, por un lado se describirán los resultados de VarCards y OpenCRAVAT, con una vertiente más puramente de anotación, y por otro GenIO y MutationDistiller, cuyo papel principal es la priorización.

Comenzando por el primer bloque de plataformas, su principal función es la de anotar las variantes que introduzcamos manualmente o a través de un fichero VCF, por lo que es sencillo hacer una comparación si nos basamos sólo y exclusivamente en la cantidad o diversidad de anotaciones. En este sentido, OpenCRAVAT supera ligeramente a VarCards con sus 156 campos de información, por los 131 que presenta ésta última, mientras que en la variedad se encuentran bastante parejos. Quizás OpenCRAVAT sea capaz de obtener anotaciones menos comunes y que sean útiles en ciertos contextos muy específicos, como el predictor CHASM o el repositorio de variantes cancerígenas CIViC (100), que pueden tener un buen aprovechamiento en experimentos de secuenciación de tumores, aunque ambos poseen un buen conjunto de anotaciones básicas. En cuanto a la rapidez y grado de uso de la plataforma hay que destacar que ambas presentan tiempos de procesamiento parecidos, en torno al minuto aproximadamente dependiendo del tamaño del *dataset* introducido, aunque OpenCRAVAT parece tener dificultades al cargar los datos en el servidor web cuando se trata de conjuntos de variantes grandes. La facilidad de uso es también una ventaja de estas herramientas, pero VarCards es incluso la más intuitiva de utilizar, con menos complejidades y ofreciendo las distintas opciones de anotación de forma clara. No obstante, OpenCRAVAT posee una característica que la diferencia del resto de plataformas como es la muestra de diferentes gráficos informativos en pantalla, pudiendo incluso construir nuestro propio *dashboard* con los gráficos que mas nos interesen; este tipo de disposiciones visuales ayudan a la interpretación de los datos y ofrecen una capa extra de funcionalidad que la diferencia del resto de plataformas de anotación.

Por otro lado, el segundo bloque de plataformas conformado por GenIO y MutationDistiller ofrece un enfoque diferente a la simple anotación de variantes, como es la priorización de éstas según una serie de factores y variables que permiten diferenciarlas entre sí. Esta priorización se basa en seleccionar de entre todas las variantes introducidas aquéllas con un mayor potencial de ser la causa de un fenotipo o enfermedad concreta, previamente introducida por el

usuario. GenIO por ejemplo nos da una lista de variantes con mayor patogenicidad según factores como el impacto de la mutación, la información de OMIM e InterVar o el predictor M-CAP, pero no nos la ofrece de forma ordenada; MutationDistiller en este caso sí nos da una lista ordenada de variantes por patogenicidad, basándose principalmente en el predictor MutationTaster con el que calcula un score para ofrecer la lista ordenada. Ambas plataformas tienen diversas opciones para introducir parámetros y filtros, como añadir el fenotipo o enfermedad que deseemos buscar de forma estandarizada y ajustar los resultados por región génica, función biológica y la rareza de la variante según la frecuencia alélica; en este sentido, MutationDistiller ofrece una gama más amplia de opciones y parámetros a introducir. En cuanto a la información presentada, GenIO da como resultado una increíble cantidad de anotaciones diversas en formato VCF, hasta 191 campos diferentes entre los que se incluyen datos sobre clasificaciones de la ACMG, algo que sin duda la hace destacar respecto al resto de herramientas de priorización. MutationDistiller por otro lado no ofrece tanta variedad de información, únicamente presenta datos sobre enfermedades relacionadas con las variantes y enlaces a diferentes repositorios para consultar información extra.

Tabla 9. Resumen con las principales ventajas e inconvenientes de las plataformas de priorización de variantes analizadas.

	Ventajas	Inconvenientes
VarCards	- Sencillez y facilidad de uso - Interfaz intuitiva y atractiva	- Menor cantidad y variedad de anotaciones
GenIO	- Gran número y variedad de anotaciones - Sencillez y facilidad de uso - Priorización de variantes	- La priorización no ofrece una lista ordenada
MutationDistiller	- Variedad de filtros y parámetros para ajustar el análisis - Lista de variantes priorizadas según score calculado	- Interfaz poco intuitiva - Tiempo de ejecución largo
OpenCRAVAT	- Gran número y variedad de anotaciones - Gráficos y visualizaciones	- Tiempo de carga de datos largo

En resumen, tal y como se muestra en la Tabla 9, es posible reunir una serie de directrices para el uso de este tipo de herramientas de priorización basándonos en sus ventajas e inconvenientes. En primer lugar, si el principal objetivo del estudio es simplemente anotar manualmente las variantes para posteriormente realizar diversos análisis u obtener información de forma cruda la mejor elección sería utilizar alguna plataforma como VarCards u OpenCRAVAT, escogiendo entre uno u otro dependiendo de la profundidad que queramos obtener y la experiencia en manejar los datos; en este sentido, VarCards es una opción más sencilla y asequible para el usuario general, mientras que OpenCRAVAT puede ser más útil para usuarios más avanzados en el campo o que quieran obtener todos los detalles posibles sobre las variantes introducidas. En caso de que el objetivo del experimento sea reducir

la lista de variantes en proyectos masivos a un pequeño conjunto donde estudiar mejor las causas genéticas de una condición la elección óptima sería el uso de plataformas como GenIO o MutationDistiller, priorizando quizás por la primera debido a su sencillez de uso, variedad y cantidad de información y apropiada priorización de variantes genéticas.

4. Conclusiones

La correcta interpretación de variantes genéticas en contextos clínicos es un proceso fundamental para dilucidar la causa de un gran número de condiciones y enfermedades de carácter hereditario, incluyendo aquellas que dependen en gran medida de variantes de mayor o menor riesgo como la mayoría de tumores. Por ello es necesario destacar que la principal conclusión de este estudio es que las herramientas de apoyo a la interpretación de variantes, ya sean predictores *in silico* o plataformas de priorización, resultan de gran ayuda en este proceso ofreciendo toda clase de información para su anotación o determinando su patogenicidad en base a métodos computacionales.

En segundo lugar, al encarar este tipo de problemas puede haber dudas en cuanto a los métodos y herramientas a elegir entre la vasta cantidad existente en la bibliografía, por lo que este trabajo puede servir de guía o punto de partida a partir del cual escoger el enfoque apropiado. La estrategia óptima en base a nuestros resultados para un caso estándar sería llevar a cabo una anotación lo más completa posible con una plataforma como VarCards, que ya de por sí nos ofrece una gran mayoría de los predictores computacionales analizados en este trabajo. Como se ha explicado anteriormente los métodos de predicción de patogenicidad que deberíamos tener en cuenta en primer lugar, en base a las métricas de evaluación analizadas, son ClinPred, BayesDel, REVEL, VEST4, fathmmMKL y PrimateAI, estudiando convenientemente las clasificaciones que ofrecen según el tipo de variantes o experimento realizado. En caso de no contar con estos predictores, como es el caso de VarCards y OpenCRAVAT, lo mejor sería obtener las predicciones mediante el repositorio dbNSFP usado en este trabajo, aunque requiere de unas mínimas capacidades de análisis bioinformático. A continuación desglosamos este punto en una serie de conclusiones claras para facilitar la tarea de elección de métodos:

- Como regla general, utilizar VarCards u OpenCRAVAT para obtener de forma rápida una serie de variantes anotadas con múltiples campos de información, tanto básicos como complejos.
- En caso de querer obtener una clasificación final de patogenicidad para un conjunto de variantes utilizar la herramienta de dbNSFP para analizarlas con los predictores ClinPred, BayesDel, REVEL, VEST4, fathmmMKL y PrimateAI.
- Para seleccionar un subconjunto de variantes con mayor potencial de ser la causa de un fenotipo en experimentos de secuenciación masivos utilizar GenIO, plataforma de priorización con la que filtrar las variantes más importantes y obtener al mismo tiempo una gran cantidad de anotaciones.

En cuanto al cumplimiento de los objetivos propuestos al comienzo del trabajo, se ha intentado seguir la organización de tareas que se propuso en el plan de trabajo inicial en la medida de lo posible, aunque la temporización varió ligeramente al finalizar la PEC 2, tal y como mencionamos anteriormente en el

apartado de planificación del trabajo. Se desplazó concretamente el bloque de análisis de las plataformas de priorización hacia la PEC 3, en lugar de para la fecha de entrega de la PEC 2 tal y como estaba planteado. El grado de cumplimiento de las tareas y objetivos propuestos podemos decir que ha sido el apropiado, tratando de obtener los hitos y resultados que se proponían, aunque algunos de ellos no los hayamos podido completar todo lo que quisiéramos, como la evaluación de las plataformas de priorización. En un inicio se planteó obtener ciertas métricas o estadísticas para comparar estas herramientas objetivamente, al igual que para los predictores *in silico*, pero como comentamos no fue posible debido a la diversidad de métodos y enfoques que presentaban; por tanto, en estas tareas concretas los resultados y discusiones correspondientes se derivan de un análisis subjetivo de las plataformas en base a ciertos factores, como ya se describió.

La planificación como se ha comentado tuvo que ser modificada sobre la marcha, por lo que se puede decir que no hubo un análisis crítico al comienzo del proyecto para calcular óptimamente los plazos y tareas que se correspondía a un trabajo como éste; quizás en un principio el enfoque de evaluación de plataformas de priorización fue muy atrevido, teniendo en cuenta que ya se haría de forma previa una comparación de predictores de patogenicidad relativamente profunda, lo que restó tiempo y dedicación al segundo bloque de análisis. Probablemente un plan de trabajo adecuado se hubiera centrado más en la evaluación de este tipo de plataformas, tan en boga en los últimos años y que carecen de comparaciones o *benchmarkings* adecuados, lo que hubiera añadido gran valor añadido a nuestro proyecto.

Finalmente, como hemos comentado este proyecto marca una línea de trabajo futura en cuanto al análisis en mayor profundidad de las diferentes plataformas de priorización existentes en la literatura, que como es lógico no se ha podido abarcar completamente en este trabajo. Los próximos estudios en relación a este tipo de herramientas deberían llenar el vacío existente en cuanto a la evaluación de herramientas de interpretación y priorización, que carecen de análisis de este tipo para que la comunidad científica y clínica tenga en cuenta las diferentes características y enfoques de cada uno de ellos para elegir correctamente cuáles utilizar o en qué condiciones. Asimismo, las líneas de trabajo futuras en relación a los predictores de patogenicidad irían en la dirección de mejorar las evaluaciones existentes, aumentando y variando los conjuntos de datos a utilizar para trabajar desde diferentes perspectivas de variantes humanas y escogiendo para el análisis nuevos clasificadores que mejoren los resultados de los ya desarrollados a día de hoy.

5. Glosario

ACMG – American College of Medical Genetics

AUC – Area Under the Curve, Área Bajo la Curva

CSV – Comma Separated Value

DoCM – Database of Curated Mutations

ESP – Exome Sequencing Project

FN – False Negative, Falso Negativo

FP – False Positive, Falso Positivo

GIAB – Genome in a Bottle

HGMD – Human Gene Mutation Database

HPO – Human Phenotype Ontology

IARC – International Agency for Research on Cancer

ICGC – International Cancer Genome Consortium

MCC – Matthews Correlation Coefficient, Coeficiente de Correlación de Matthews

MKL – Multiple Kernel Library

NCBI – National Center for Biotechnology Information

NGS – Next Generation Sequencing, Secuenciación de Próxima Generación

NPV – Negative Predictive Value, Valor Predictivo Negativo

OMIM – Online Mendelian Inheritance in Man

PEC – Prueba de Evaluación Continua

PPV – Positive Predictive Value, Valor Predictivo Positivo

ROC – Receiver Operating Characteristic

SVM – Support Vector Machine, Máquina de Vector de Soporte

TN – True Negative, Verdadero Negativo

TP – True Positive, Verdadero Positivo

VCF – Variant Call Format

VEP – Variant Effect Predictor

VUS – Variant of Uncertain Significance, Variante de Significado Incierto

6. Bibliografía

1. Eilbeck K, Quinlan A, Yandell M. Settling the score: variant prioritization and Mendelian disease. Nat Rev Genet. octubre de 2017;18(10):599-612.

2. McLaren W, Gil L, Hunt SE, Riat HS, Ritchie GRS, Thormann A, et al. The Ensembl Variant Effect Predictor. Genome Biol. 2016;17(1):1-14.

3. Yang H, Wang K. Genomic variant annotation and prioritization with ANNOVAR and wANNOVAR. Nat Protoc. 2015;10(10):1556-66.

4. Cingolani P, Platts A, Wang LL, Coon M, Nguyen T, Wang L, et al. A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of Drosophila melanogaster strain w1118; iso-2; iso-3. Fly (Austin). 2012;6(2):80-92.

5. Grimm DG, Azencott CA, Aicheler F, Gieraths U, Macarthur DG, Samocha KE, et al. Prioritization of variants detected by next generation sequencing according to the mutation tolerance and mutational architecture of the corresponding genes. BMC Genomics. 2019;102(1):2125-37.

6. Calculated consequences [Internet]. [citado 6 de diciembre de 2020]. Disponible en: https://m.ensembl.org/info/genome/variation/prediction/predicted_data.html

7. Fernald GH, Capriotti E, Daneshjou R, Karczewski KJ, Altman RB. Bioinformatics challenges for personalized medicine. Bioinformatics. 2011;27(13):1741-8.

8. Nishizaki SS, Boyle AP. Mining the Unknown: Assigning Function to Noncoding Single Nucleotide Polymorphisms. Trends Genet. 2017;33(1):34-45.

9. Yang Y, Muzny DM, Reid JG, Bainbridge MN, Willis A, Ward PA, et al. Clinical whole-exome sequencing for the diagnosis of mendelian disorders. N Engl J Med. 17 de octubre de 2013;369(16):1502-11.

10. Thusberg J, Olatubosun A, Vihinen M. Performance of mutation pathogenicity prediction methods on missense variants. Hum Mutat. 2011;32(4):358-68.

11. Gnad F, Baucom A, Mukhyala K, Manning G, Zhang Z. Assessment of computational methods for predicting the effects of missense mutations in human cancers. BMC Genomics. 2013;14 Suppl 3(Suppl 3):S7.

12. Frousios K, Iliopoulos CS, Schlitt T, Simpson MA. Predicting the functional consequences of non-synonymous DNA sequence variants - evaluation of bioinformatics tools and development of a consensus strategy. Genomics. 2013;102(4):223-8.

13. Dong C, Wei P, Jian X, Gibbs R, Boerwinkle E, Wang K, et al. Comparison and integration of deleteriousness prediction methods for nonsynonymous SNVs in whole exome sequencing studies. Hum Mol Genet. 2015;24(8):2125-37.

14. Li J, Zhao T, Zhang Y, Zhang K, Shi L, Chen Y, et al. Performance evaluation of pathogenicity-computation methods for missense variants. Nucleic Acids Res. 2018;46(15):7793-804.

15. Hassan MS, Shaalan AA, Dessouky MI, Abdelnaiem AE, ElHefnawi M. Evaluation of computational techniques for predicting non-synonymous single nucleotide variants pathogenicity. Genomics. 2019;111(4):869-82.

16. Ghosh R, Oak N, Plon SE. Evaluation of in silico algorithms for use with ACMG/AMP clinical variant interpretation guidelines. Genome Biol. 2017;18(1):1-12.

17. Pengelly RJ, Alom T, Zhang Z, Hunt D, Ennis S, Collins A. Evaluating phenotype-driven approaches for genetic diagnoses from exomes in a clinical setting. Sci Rep. 2017;7(1):1-7.

18. gnu.org [Internet]. [citado 6 de diciembre de 2020]. Disponible en: https://www.gnu.org/software/bash/

19. The GNU Awk User's Guide [Internet]. [citado 6 de diciembre de 2020]. Disponible en: https://www.gnu.org/software/gawk/manual/gawk.html

20. Khan MRA, Brandenburger T. ROCit: Performance Assessment of Binary Classifier with Visualization [Internet]. 2020 [citado 6 de diciembre de 2020]. Disponible en: https://CRAN.R-project.org/package=ROCit

21. Kuhn M. The caret Package [Internet]. [citado 6 de diciembre de 2020]. Disponible en: http://topepo.github.io/caret/index.html

22. ClinVar [Internet]. [citado 7 de diciembre de 2020]. Disponible en: https://www.ncbi.nlm.nih.gov/clinvar/

23. Karczewski KJ, Francioli LC, Tiao G, Cummings BB, Alföldi J, Wang Q, et al. The mutational constraint spectrum quantified from variation in 141,456 humans. Nature. mayo de 2020;581(7809):434-43.

24. Zook JM, Catoe D, McDaniel J, Vang L, Spies N, Sidow A, et al. Extensive sequencing of seven human genomes to characterize benchmark reference materials. Sci Data. 7 de junio de 2016;3(1):160025.

25. s.r.o BS. GanttProject: free project management tool for Windows, macOS and Linux [Internet]. GanttProject. [citado 7 de diciembre de 2020]. Disponible en: https://www.ganttproject.biz

26. Liu X, Wu C, Li C, Boerwinkle E. dbNSFP v3.0: A One-Stop Database of Functional Predictions and Annotations for Human Nonsynonymous and Splice-Site SNVs. Vol. 37, Human Mutation. 2016. 235-241 p.

27. Java | Oracle [Internet]. [citado 8 de diciembre de 2020]. Disponible en: https://www.java.com/es/

28. microsoft/WSL2-Linux-Kernel [Internet]. Microsoft; 2020 [citado 8 de diciembre de 2020]. Disponible en: https://github.com/microsoft/WSL2-Linux-Kernel

29. rstudio/rmarkdown [Internet]. RStudio; 2020 [citado 8 de diciembre de 2020]. Disponible en: https://github.com/rstudio/rmarkdown

30. R: The R Project for Statistical Computing [Internet]. [citado 8 de diciembre de 2020]. Disponible en: https://www.r-project.org/

31. RStudio | Open source & professional software for data science teams [Internet]. [citado 8 de diciembre de 2020]. Disponible en: https://rstudio.com/

32. Gorman B. mltools: Machine Learning Tools [Internet]. 2018 [citado 8 de diciembre de 2020]. Disponible en: https://CRAN.R-project.org/package=mltools

33. Xie [aut Y, cre, Vogt A, Andrew A, Zvoleff A, http://www.andre-simon.de) AS (the C files under inst/themes/ were derived from the H package, et al. knitr: A General-Purpose Package for Dynamic Report Generation in R [Internet]. 2020 [citado 8 de diciembre de 2020]. Disponible en: https://CRAN.R-project.org/package=knitr

34. Zhu [aut H, cre, Travison T, Tsai T, Beasley W, Xie Y, et al. kableExtra: Construct Complex Table with «kable» and Pipe Syntax [Internet]. 2020 [citado 8 de diciembre de 2020]. Disponible en: https://CRAN.R-project.org/package=kableExtra

35. Wickham H, Chang W, Henry L, Pedersen TL, Takahashi K, Wilke C, et al. ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics [Internet]. 2020 [citado 8 de diciembre de 2020]. Disponible en: https://CRAN.R-project.org/package=ggplot2

36. Wickham H. reshape: Flexibly Reshape Data [Internet]. 2018 [citado 8 de diciembre de 2020]. Disponible en: https://CRAN.R-project.org/package=reshape

37. Ng PC, Henikoff S. Accounting for human polymorphisms predicted to affect protein function. Genome Res. 2002;12(3):436-46.

38. Wong WC, Kim D, Carter H, Diekhans M, Ryan MC, Karchin R. CHASM and SNVBox: Toolkit for detecting biologically important single nucleotide mutations in cancer. Bioinformatics. 2011;27(15):2147-8.

39. Stenson PD, Ball EV, Mort M, Phillips AD, Shiel JA, Thomas NST, et al. Human Gene Mutation Database (HGMD): 2003 update. Hum Mutat. junio de 2003;21(6):577-81.

40. Exome Variant Server [Internet]. [citado 8 de diciembre de 2020]. Disponible en: https://evs.gs.washington.edu/EVS/

41. Carter H, Douville C, Stenson PD, Cooper DN, Karchin R. Identifying Mendelian disease genes with the variant effect scoring tool. BMC Genomics. 2013;14 Suppl 3(Suppl 3):S3.

42. Raimondi D, Tanyalcin I, FertCrossed JSD, Gazzo A, Orlando G, Lenaerts T, et al. DEOGEN2: Prediction and interactive visualization of single amino acid variant deleteriousness in human proteins. Nucleic Acids Res. 2017;45(W1):W201-6.

43. Malhis N, Jacobson M, Jones SJM, Gsponer J. LIST-S2: taxonomy based sorting of deleterious missense mutations across species. Nucleic Acids Res. 2020;48(W1):W154-61.

44. Sundaram L, Gao H, Padigepati SR, McRae JF, Li Y, Kosmicki JA, et al. Predicting the clinical impact of human mutation with deep neural networks. Nat Genet. 1 de agosto de 2018;50(8):1161-70.

45. Adzhubei I, Jordan DM, Sunyaev SR. Predicting functional effect of human missense mutations using PolyPhen-2. Current Protocols in Human Genetics. 2013. 1-41 p.

46. Siepel A, Pollard KS, Haussler D. New methods for detecting lineage-specific selection. Lect Notes Comput Sci Subser Lect Notes Artif Intell Lect Notes Bioinforma. 2006;3909 LNBI:190-205.

47. Rentzsch P, Witten D, Cooper GM, Shendure J, Kircher M. CADD: Predicting the deleteriousness of variants throughout the human genome. Nucleic Acids Res. 2019;47(D1):D886-94.

48. Quang D, Chen Y, Xie X. DANN: A deep learning approach for annotating the pathogenicity of genetic variants. Bioinformatics. 2015;31(5):761-3.

49. Shihab HA, Rogers MF, Gough J, Mort M, Cooper DN, Day INM, et al. An integrative approach to predicting the functional effects of non-coding and coding sequence variation. Bioinformatics. 2015;31(10):1536-43.

50. Schwarz JM, Rödelsperger C, Schuelke M, Seelow D. MutationTaster evaluates disease-causing potential of sequence alterations. Nat Methods. 2010;7(8):575-6.

51. Liu X, Wu C, Li C, Boerwinkle E. dbNSFP v3.0: A One-Stop Database of Functional Predictions and Annotations for Human Nonsynonymous and Splice-Site SNVs. Vol. 37, Human Mutation. 2016. p. 235-41.

52. Pejaver V, Urresti J, Lugo-Martinez J, Pagel K, Lin GN, Nam H-J, et al. MutPred2: inferring the molecular and phenotypic impact of amino acid variants. bioRxiv. 2017;134981.

53. Choi Y, Sims GE, Murphy S, Miller JR, Chan AP. Predicting the Functional Effect of Amino Acid Substitutions and Indels. PLOS ONE. 8 de octubre de 2012;7(10):e46688.

54. Reva B, Antipin Y, Sander C. Predicting the functional impact of protein mutations: Application to cancer genomics. Nucleic Acids Res. 2011;39(17):37-43.

55. Chun S, Fay JC. Identification of deleterious mutations within three human genomes. Genome Res. 2009;19(9):1553-61.

56. Davydov E V., Goode DL, Sirota M, Cooper GM, Sidow A, Batzoglou S. Identifying a high fraction of the human genome to be under selective constraint using GERP++. PLoS Comput Biol. 2010;6(12).

57. Identifying novel constrained elements by exploiting biased substitution patterns | Bioinformatics | Oxford Academic [Internet]. [citado 9 de diciembre de 2020]. Disponible en: https://academic.oup.com/bioinformatics/article/25/12/i54/187307

58. Siepel A, Bejerano G, Pedersen JS, Hinrichs AS, Hou M, Rosenbloom K, et al. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. Genome Res. 8 de enero de 2005;15(8):1034-50.

59. Ioannidis NM, Rothstein JH, Pejaver V, Middha S, McDonnell SK, Baheti S, et al. REVEL: An Ensemble Method for Predicting the Pathogenicity of Rare Missense Variants. Am J Hum Genet. 2016;99(4):877-85.

60. Ionita-Laza I, Mccallum K, Xu B, Buxbaum JD. A spectral approach integrating functional genomic annotations for coding and noncoding variants. Nat Genet. 2016;48(2):214-20.

61. Feng BJ. PERCH: A Unified Framework for Disease Gene Prioritization. Hum Mutat. 2017;38(3):243-51.

62. Gulko B, Hubisz MJ, Gronau I, Siepel A. Probabilities of Fitness Consequences for Point Mutations Across the Human Genome. Nat Genet. marzo de 2015;47(3):276-83.

63. Alirezaie N, Kernohan KD, Hartley T, Majewski J, Hocking TD. ClinPred: Prediction Tool to Identify Disease-Relevant Nonsynonymous Single-Nucleotide Variants. Am J Hum Genet. 2018;103(4):474-83.

64. Sarkar A, Yang Y, Vihinen M. Variation benchmark datasets: update, criteria, quality and applications. Database J Biol Databases Curation. 2020;2020:1-16.

65. Tian Y, Pesaran T, Chamberlin A, Fenwick RB, Li S, Gau CL, et al. REVEL and BayesDel outperform other in silico meta-predictors for clinical variant classification. Sci Rep. 2019;9(1):1-6.

66. Welcome - IARC TP53 Database [Internet]. [citado 10 de diciembre de 2020]. Disponible en: https://p53.iarc.fr/

67. International Cancer Genome Consortium [Internet]. [citado 10 de diciembre de 2020]. Disponible en: https://icgc.org/

68. Home - SNP - NCBI [Internet]. [citado 10 de diciembre de 2020]. Disponible en: https://www.ncbi.nlm.nih.gov/snp/

69. VariBench [Internet]. [citado 10 de diciembre de 2020]. Disponible en: http://structure.bmc.lu.se/VariBench/

70. UniProt [Internet]. [citado 10 de diciembre de 2020]. Disponible en: https://www.uniprot.org/

71. IGSR | samples [Internet]. [citado 10 de diciembre de 2020]. Disponible en: https://www.internationalgenome.org/data-portal/sample

72. gnomAD [Internet]. [citado 10 de diciembre de 2020]. Disponible en: https://gnomad.broadinstitute.org/

73. HGMD® home page [Internet]. [citado 10 de diciembre de 2020]. Disponible en: http://www.hgmd.cf.ac.uk/ac/index.php

74. PhenCode: Paving the Path between Phenotype and Genome [Internet]. [citado 10 de diciembre de 2020]. Disponible en: http://phencode.bx.psu.edu/

75. UCSC Genome Browser Home [Internet]. [citado 10 de diciembre de 2020]. Disponible en: http://genome.ucsc.edu/

76. DoCM - Database of Curated Mutations [Internet]. [citado 10 de diciembre de 2020]. Disponible en: http://docm.info/

77. justin.zook@nist.gov. Genome in a Bottle [Internet]. NIST. 2012 [citado 10 de diciembre de 2020]. Disponible en: https://www.nist.gov/programs-projects/genome-bottle

78. samtools/hts-specs [Internet]. samtools; 2020 [citado 22 de diciembre de 2020]. Disponible en: https://github.com/samtools/hts-specs

79. Kato S, Han S-Y, Liu W, Otsuka K, Shibata H, Kanamaru R, et al. Understanding the function-structure and function-mutation relationships of p53 tumor suppressor protein by high-resolution missense mutation analysis. Proc Natl Acad Sci U S A. 8 de julio de 2003;100(14):8424-9.

80. Bouaoun L, Sonkin D, Ardin M, Hollstein M, Byrnes G, Zavadil J, et al. TP53 Variations in Human Cancers: New Lessons from the IARC TP53 Database and Genomics Data. Hum Mutat. 2016;37(9):865-76.

81. Vihinen M. How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis. BMC Genomics. 2012;13 Suppl 4(Suppl 4).

82. Li J, Shi L, Zhang K, Zhang Y, Hu S, Zhao T, et al. VarCards: An integrated genetic and clinical database for coding variants in the human genome. Nucleic Acids Res. 2018;46(D1):D1039-48.

83. Koile D, Cordoba M, de Sousa Serro M, Kauffman MA, Yankilevich P. GenIO: A phenotype-genotype analysis web server for clinical genomics of rare diseases. BMC Bioinformatics. 2018;19(1):1-6.

84. Hombach D, Schuelke M, Knierim E, Ehmke N, Schwarz JM, Fischer-Zirnsak B, et al. MutationDistiller: User-driven identification of pathogenic DNA variants. Nucleic Acids Res. 2019;47(W1):W114-20.

85. Pagel KA, Kim R, Moad K, Busby B, Zheng L, Tokheim C, et al. Integrated Informatics Analysis of Cancer-Related Variants. JCO Clin Cancer Inform. 2020;(4):310-7.

86. Wang K, Li M, Hakonarson H. ANNOVAR: Functional annotation of genetic variants from high-throughput sequencing data. Nucleic Acids Res. 3 de julio de 2010;38(16).

87. McLaren W, Gil L, Hunt SE, Riat HS, Ritchie GRS, Thormann A, et al. The Ensembl Variant Effect Predictor. Genome Biol. 2016;17(1):1-14.

88. ClinVar [Internet]. Disponible en: https://www.ncbi.nlm.nih.gov/clinvar/

89. Li Q, Wang K. InterVar: Clinical Interpretation of Genetic Variants by the 2015 ACMG-AMP Guidelines. Am J Hum Genet. 2017;100(2):267-80.

90. Jagadeesh KA, Wenger AM, Berger MJ, Guturu H, Stenson PD, Cooper DN, et al. M-CAP eliminates a majority of variants of uncertain significance in clinical exomes at high sensitivity. Nat Genet. 2016;48(12):1581-6.

91. Schwarz JM, Rödelsperger C, Schuelke M, Seelow D. MutationTaster evaluates disease-causing potential of sequence alterations. Nat Methods. 2010;7(8):575-6.

92. Wong WC, Kim D, Carter H, Diekhans M, Ryan MC, Karchin R. CHASM and SNVBox: Toolkit for detecting biologically important single nucleotide mutations in cancer. Bioinformatics. 2011;27(15):2147-8.

93. VarCards [Internet]. [citado 20 de diciembre de 2020]. Disponible en: http://159.226.67.237/sun/varcards/welcome

94. GenIO [Internet]. [citado 20 de diciembre de 2020]. Disponible en: https://bioinformatics.ibioba-mpsp-conicet.gov.ar/GenIO

95. Richards S, Aziz N, Bale S, Bick D, Das S, Gastier-Foster J, et al. Standards and guidelines for the interpretation of sequence variants: A joint consensus recommendation of the American College of Medical Genetics and Genomics and the Association for Molecular Pathology. Genet Med. 8 de mayo de 2015;17(5):405-24.

96. MutationDistiller [Internet]. [citado 20 de diciembre de 2020]. Disponible en: https://www.mutationdistiller.org/

97. Kanehisa M, Furumichi M, Sato Y, Ishiguro-Watanabe M, Tanabe M. KEGG: integrating viruses and cellular organisms. Nucleic Acids Res. 30 de octubre de 2020;gkaa970.

98. Jassal B, Matthews L, Viteri G, Gong C, Lorente P, Fabregat A, et al. The reactome pathway knowledgebase. Nucleic Acids Res. 8 de enero de 2020;48(D1):D498-503.

99. Home - Reactome Pathway Database [Internet]. [citado 21 de diciembre de 2020]. Disponible en: https://reactome.org/

100. Griffith M, Spies NC, Krysiak K, McMichael JF, Coffman AC, Danos AM, et al. CIViC is a community knowledgebase for expert crowdsourcing the clinical interpretation of variants in cancer. Nat Genet. febrero de 2017;49(2):170-4.

7. Anexos

Junto a este documento se adjuntarán los siguientes ficheros anexos:

- Código en formato R Markdown de los análisis realizados para la evaluación de predictores de patogenicidad del conjunto de datos de ClinVar, además del correspondiente HTML exportado.
- Código en formato R Markdown de los análisis realizados para la evaluación de predictores de patogenicidad del conjunto de datos de IARC/ICGC, además del correspondiente HTML exportado.
- Código en formato R Markdown de los análisis realizados para la evaluación de predictores de patogenicidad del conjunto de datos de GIAB, además del correspondiente HTML exportado.
- Predicciones de los diferentes métodos para cada una de las variantes del conjunto de datos de ClinVar en formato CSV.
- Predicciones de los diferentes métodos para cada una de las variantes del conjunto de datos de IARC/ICGC en formato CSV.
- Predicciones de los diferentes métodos para cada una de las variantes del conjunto de datos de GIAB en formato CSV.
- Fichero en formato TSV con las anotaciones obtenidas mediante la plataforma VarCards para las variantes patógenas de ClinVar.
- Fichero en formato TSV con las anotaciones obtenidas mediante la plataforma VarCards para las variantes benignas de ClinVar.
- Fichero en formato TSV con las anotaciones obtenidas mediante la plataforma VarCards para las variantes de GIAB.
- Fichero en formato CSV con las anotaciones obtenidas mediante la plataforma MutationDistiller para las variantes priorizadas de GIAB.
- Fichero en formato VCF con las anotaciones obtenidas mediante la plataforma GenIO para todas las variantes de GIAB.
- Fichero en formato TSV con las anotaciones obtenidas mediante la plataforma OpenCRAVAT para las variantes patógenas de ClinVar.
- Fichero en formato TSV con las anotaciones obtenidas mediante la plataforma OpenCRAVAT para las variantes benignas de ClinVar.
- Fichero en formato TSV con las anotaciones obtenidas mediante la plataforma OpenCRAVAT para las variantes de GIAB.

7.1 Código en formato R Markdown de los análisis realizados para la evaluación de predictores de patogenicidad del conjunto de datos de ClinVar

```
---
title: "Evaluación predictores de patogenicidad con ClinVar"
author: "Víctor Manuel Duarte Rute"
date: "2/11/2020"
output:
  html_document:
    df_print: paged
---
```

<style type="text/css">
.main-container {
  max-width: 1800px;
  margin-left: auto;
  margin-right: auto;
}
</style>

```{r setup, include=FALSE, warning=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(caret)
library(pROC)
library(ROCit)
library(mltools)
library(knitr)
library(kableExtra)
library(ggplot2)
library(reshape)
```

En este informe basado en R Markdown vamos a ir realizando los diferentes procesamientos de los datos y los análisis pertinentes para evaluar el rendimiento de los predictores de patogenicidad elegidos. Para ello partimos de un fichero tabulado procedente de la anotación de las variantes con el repositorio dbNSFP, que incluye multitud de información interesante para caracterizar todo tipo de variantes, para obtener las predicciones de patogenicidad de las variantes presentes en nuestro dataset. Para maximizar la precisión de los análisis se han llevado a cabo preprocesamientos y filtros previos para quedarnos con aquellas variantes más indicadas para este tipo de algoritmos; más concretamente, nos quedamos con variantes con una consecuencia de 'missense_variant' catalogada por Ensembl, ya que son aquéllas que provocan un cambio de aminoácido y por tanto las que más interesan por su potencial clínico. Asímismo, nos quedamos ocn variantes de tipo SNV, o Single Nucleotide Variation, mucho más comunes y sencillas de analizar por este tipo de predictores in silico.

```{r}
setwd("D:/MÁSTER BIOESTADÍSTICA UOC/TFM/files (new analysis)/RMarkdowns_reports")
```

Cargamos los datos tabulados en sendas tablas, una para las clasificadas como patógenas y otra para las benignas.

```{r}
clinvar_pathogenic <- read.csv("clinvar_reviewed_SNVs_missense_pathogenic_dbNSFP.out", header = TRUE, sep = "\t")
clinvar_benign <- read.csv("clinvar_reviewed_SNVs_missense_benign_dbNSFP.out", header = TRUE, sep = "\t")
```

Creamos la primera función encargada de reformatear algunos de los campos de la tabla, concretamente aquellos predictores que poseen múltiples valores debido a la presencia de diferentes transcritos. La aplicación de esta función hace que se haga un merge de todas las clasificaciones para cada variante y nos quedemos con la etiqueta mayoritaria, sea patógena (D, o Deleterious, a partir de ahora etiquetada como '1') o benigna (T, o Tolerated, a partir de ahora etiquetada como '0').

```{r}
reformat_predictions <- function(predictions) {
  predictions_reformat <- c()
  for (i in 1:length(predictions)) {
    split_predictions <- strsplit(predictions[i], ";")
    tolerated <- 0
    deleterious <- 0
    for (j in 1:lengths(split_predictions)) {
      if (split_predictions[[1]][j] == "T") {
        tolerated <- tolerated + 1
        }
      if (split_predictions[[1]][j] == "D") {
        deleterious <- deleterious + 1
        }
    }
```

```
    if (deleterious > tolerated)
      predictions_reformat[i] = 1
    if (deleterious < tolerated)
      predictions_reformat[i] = 0
    if (tolerated == 0 && deleterious == 0)
      predictions_reformat[i] = "-"
    if (tolerated == deleterious && tolerated != 0)
      predictions_reformat[i] = 1
  }

  return(predictions_reformat)
}
```

Aplicamos dicha función a los 3 predictores que poseen esta clasificación múltiple, como son SIFT, DEOGEN2 y LIST S2.

```{r}
clinvar_pathogenic$SIFT_pred <- reformat_predictions(clinvar_pathogenic$SIFT_pred)
clinvar_pathogenic$DEOGEN2_pred <- reformat_predictions(clinvar_pathogenic$DEOGEN2_pred)
clinvar_pathogenic$LIST.S2_pred <- reformat_predictions(clinvar_pathogenic$LIST.S2_pred)

clinvar_benign$SIFT_pred <- reformat_predictions(clinvar_benign$SIFT_pred)
clinvar_benign$DEOGEN2_pred <- reformat_predictions(clinvar_benign$DEOGEN2_pred)
clinvar_benign$LIST.S2_pred <- reformat_predictions(clinvar_benign$LIST.S2_pred)
```

La siguiente función que creamos se encarga de reformatear la gran mayoría de los predictores para tener la clasificación numérica que hemos comentado anteriormente, aunque en este caso es más sencillo al no ser valores múltiples.

```{r}
reformat_points <- function(predictions) {
  for (i in 1:length(predictions)) {
    if (predictions[i] == ".") {
      predictions[i] <- "-"
    }
    if (predictions[i] == "D") {
      predictions[i] <- 1
    }
    if (predictions[i] == "T" || predictions[i] == "N") {
      predictions[i] <- 0
    }
  }

  return(predictions)
}
```

Aplicamos la función anterior con los predictores MetaLR, MetaSVM, PrimateAI, fathmm_MKL, BayesDel y ClinPred.

```{r}
clinvar_pathogenic$MetaLR_pred <- reformat_points(clinvar_pathogenic$MetaLR_pred)
clinvar_pathogenic$MetaSVM_pred <- reformat_points(clinvar_pathogenic$MetaSVM_pred)
clinvar_pathogenic$PrimateAI_pred <- reformat_points(clinvar_pathogenic$PrimateAI_pred)
clinvar_pathogenic$fathmm.MKL_coding_pred <-
reformat_points(clinvar_pathogenic$fathmm.MKL_coding_pred)
clinvar_pathogenic$BayesDel_addAF_pred <-
reformat_points(clinvar_pathogenic$BayesDel_addAF_pred)
clinvar_pathogenic$ClinPred_pred <- reformat_points(clinvar_pathogenic$ClinPred_pred)

clinvar_benign$MetaLR_pred <- reformat_points(clinvar_benign$MetaLR_pred)
clinvar_benign$MetaSVM_pred <- reformat_points(clinvar_benign$MetaSVM_pred)
clinvar_benign$PrimateAI_pred <- reformat_points(clinvar_benign$PrimateAI_pred)
clinvar_benign$fathmm.MKL_coding_pred <- reformat_points(clinvar_benign$fathmm.MKL_coding_pred)
```

```
clinvar_benign$BayesDel_addAF_pred <- reformat_points(clinvar_benign$BayesDel_addAF_pred)
clinvar_benign$ClinPred_pred <- reformat_points(clinvar_benign$ClinPred_pred)
```

A continuación los siguientes bloques de código se encargan de reformatear el resto de predictores a analizar, cuya peculiaridad es que no poseen de entrada una clasificación categórica, por lo que es necesario que nosotros mismos convirtamos la predicción numérica (cada predictor con su propio rango de valores) en una clasificación basada en etiqueta. Esta conversión se realiza en base a las propias estimaciones de los desarrolladores de los predictores, que proponen un cutoff o valor determinado a partir del cual establecer la división. Cada bloque por tanto se dirige a la transformación de un predictor distinto según su propio rango de valores.

VEST4

```{r}
for (i in 1:length(clinvar_pathogenic$VEST4_score)) {
  split_scores <- strsplit(clinvar_pathogenic$VEST4_score[i], ";")
  highest_score <- "NA"
  boolean <- 0
  for (j in 1:lengths(split_scores)) {
    if (grepl("^[0-9]", split_scores[[1]][j])) {
      score <- as.numeric(split_scores[[1]][j])
      if (boolean == 0) {
        highest_score <- score
        boolean <- 1
      }
      if (score > highest_score) {
        highest_score <- score
      }
    }
  }
  if (highest_score == "NA") {
    clinvar_pathogenic$VEST4_pred[i] <- "-"
  } else if (highest_score >= 0.5) {
    clinvar_pathogenic$VEST4_pred[i] <- 1
  } else if (highest_score < 0.5) {
    clinvar_pathogenic$VEST4_pred[i] <- 0
  }
}
```

REVEL

```{r}
for (i in 1:length(clinvar_pathogenic$REVEL_score)) {
  split_scores <- strsplit(clinvar_pathogenic$REVEL_score[i], ";")
  highest_score <- "NA"
  boolean <- 0
  for (j in 1:lengths(split_scores)) {
    if (grepl("^[0-9]", split_scores[[1]][j])) {
      score <- as.numeric(split_scores[[1]][j])
      if (boolean == 0) {
        highest_score <- score
        boolean <- 1
      }
      if (score > highest_score) {
        highest_score <- score
      }
    }
  }
  if (highest_score == "NA") {
    clinvar_pathogenic$REVEL_pred[i] <- "-"
  } else if (highest_score >= 0.4) {
    clinvar_pathogenic$REVEL_pred[i] <- 1
  } else if (highest_score < 0.4) {
    clinvar_pathogenic$REVEL_pred[i] <- 0
  }
```

```
}
```

CADD

```{r}
for (i in 1:length(clinvar_pathogenic$CADD_phred_hg19)) {
  if (clinvar_pathogenic$CADD_phred_hg19[i] == ".") {
    clinvar_pathogenic$CADD_hg19_pred[i] <- "-"
  } else if (clinvar_pathogenic$CADD_phred_hg19[i] > 20) {
    clinvar_pathogenic$CADD_hg19_pred[i] <- 1
  } else if (clinvar_pathogenic$CADD_phred_hg19[i] <= 20) {
    clinvar_pathogenic$CADD_hg19_pred[i] <- 0
  }
}
```

DANN

```{r}
for (i in 1:length(clinvar_pathogenic$DANN_score)) {
  if (clinvar_pathogenic$DANN_score[i] == ".") {
    clinvar_pathogenic$DANN_pred[i] <- "-"
  } else if (clinvar_pathogenic$DANN_score[i] >= 0.99) {
    clinvar_pathogenic$DANN_pred[i] <- 1
  } else if (clinvar_pathogenic$DANN_score[i] < 0.99) {
    clinvar_pathogenic$DANN_pred[i] <- 0
  }
}
```

Eigen

```{r}
for (i in 1:length(clinvar_pathogenic$Eigen.raw_coding)) {
  if (clinvar_pathogenic$Eigen.raw_coding[i] == ".") {
    clinvar_pathogenic$Eigen_pred[i] <- "-"
  } else if (clinvar_pathogenic$Eigen.raw_coding[i] >= 0) {
    clinvar_pathogenic$Eigen_pred[i] <- 1
  } else if (clinvar_pathogenic$Eigen.raw_coding[i] < 0) {
    clinvar_pathogenic$Eigen_pred[i] <- 0
  }
}
```

GERP

```{r}
for (i in 1:length(clinvar_pathogenic$GERP.._RS)) {
  if (clinvar_pathogenic$GERP.._RS[i] == ".") {
    clinvar_pathogenic$GERP_pred[i] <- "-"
  } else if (clinvar_pathogenic$GERP.._RS[i] >= 2) {
    clinvar_pathogenic$GERP_pred[i] <- 1
  } else if (clinvar_pathogenic$GERP.._RS[i] < 2) {
    clinvar_pathogenic$GERP_pred[i] <- 0
  }
}
```

PhyloP

```{r}
for (i in 1:length(clinvar_pathogenic$phyloP100way_vertebrate)) {
  if (clinvar_pathogenic$phyloP100way_vertebrate[i] == ".") {
    clinvar_pathogenic$phyloP100way_vertebrate_pred[i] <- "-"
  } else if (clinvar_pathogenic$phyloP100way_vertebrate[i] > 2) {
    clinvar_pathogenic$phyloP100way_vertebrate_pred[i] <- 1
```

```
  } else if (clinvar_pathogenic$phyloP100way_vertebrate[i] < 2) {
    clinvar_pathogenic$phyloP100way_vertebrate_pred[i] <- 0
  }
}
```

Finalmente creamos la columna de clasificación, en la que se especifica el valor real de la patogenicidad de la variante según las características del conjunto de datos.

```{r}
clinvar_pathogenic$classification <- 1
```

A continuación se lleva a cabo la misma transformación anterior con el conjunto de datos de variantes benignas.

VEST4

```{r}
for (i in 1:length(clinvar_benign$VEST4_score)) {
  split_scores <- strsplit(clinvar_benign$VEST4_score[i], ";")
  highest_score <- "NA"
  boolean <- 0
  for (j in 1:lengths(split_scores)) {
    if (grepl("^[0-9]", split_scores[[1]][j])) {
      score <- as.numeric(split_scores[[1]][j])
      if (boolean == 0) {
        highest_score <- score
        boolean <- 1
      }
      if (score > highest_score) {
        highest_score <- score
      }
    }
  }
  if (highest_score == "NA") {
    clinvar_benign$VEST4_pred[i] <- "-"
  } else if (highest_score >= 0.5) {
    clinvar_benign$VEST4_pred[i] <- 1
  } else if (highest_score < 0.5) {
    clinvar_benign$VEST4_pred[i] <- 0
  }
}
```

REVEL

```{r}
for (i in 1:length(clinvar_benign$REVEL_score)) {
  split_scores <- strsplit(clinvar_benign$REVEL_score[i], ";")
  highest_score <- "NA"
  boolean <- 0
  for (j in 1:lengths(split_scores)) {
    if (grepl("^[0-9]", split_scores[[1]][j])) {
      score <- as.numeric(split_scores[[1]][j])
      if (boolean == 0) {
        highest_score <- score
        boolean <- 1
      }
      if (score > highest_score) {
        highest_score <- score
      }
    }
  }
  if (highest_score == "NA") {
    clinvar_benign$REVEL_pred[i] <- "-"
  } else if (highest_score >= 0.4) {
```

```
    clinvar_benign$REVEL_pred[i] <- 1
  } else if (highest_score < 0.4) {
    clinvar_benign$REVEL_pred[i] <- 0
  }
}
```

CADD

```{r}
for (i in 1:length(clinvar_benign$CADD_phred_hg19)) {
  if (clinvar_benign$CADD_phred_hg19[i] == ".") {
    clinvar_benign$CADD_hg19_pred[i] <- "-"
  } else if (clinvar_benign$CADD_phred_hg19[i] > 20) {
    clinvar_benign$CADD_hg19_pred[i] <- 1
  } else if (clinvar_benign$CADD_phred_hg19[i] <= 20) {
    clinvar_benign$CADD_hg19_pred[i] <- 0
  }
}
```

DANN

```{r}
for (i in 1:length(clinvar_benign$DANN_score)) {
  if (clinvar_benign$DANN_score[i] == ".") {
    clinvar_benign$DANN_pred[i] <- "-"
  } else if (clinvar_benign$DANN_score[i] >= 0.99) {
    clinvar_benign$DANN_pred[i] <- 1
  } else if (clinvar_benign$DANN_score[i] < 0.99) {
    clinvar_benign$DANN_pred[i] <- 0
  }
}
```

Eigen

```{r}
for (i in 1:length(clinvar_benign$Eigen.raw_coding)) {
  if (clinvar_benign$Eigen.raw_coding[i] == ".") {
    clinvar_benign$Eigen_pred[i] <- "-"
  } else if (clinvar_benign$Eigen.raw_coding[i] >= 0) {
    clinvar_benign$Eigen_pred[i] <- 1
  } else if (clinvar_benign$Eigen.raw_coding[i] < 0) {
    clinvar_benign$Eigen_pred[i] <- 0
  }
}
```

GERP

```{r}
for (i in 1:length(clinvar_benign$GERP.._RS)) {
  if (clinvar_benign$GERP.._RS[i] == ".") {
    clinvar_benign$GERP_pred[i] <- "-"
  } else if (clinvar_benign$GERP.._RS[i] >= 2) {
    clinvar_benign$GERP_pred[i] <- 1
  } else if (clinvar_benign$GERP.._RS[i] < 2) {
    clinvar_benign$GERP_pred[i] <- 0
  }
}
```

PhyloP

```{r}
for (i in 1:length(clinvar_benign$phyloP100way_vertebrate)) {
```

```
  if (clinvar_benign$phyloP100way_vertebrate[i] == ".") {
    clinvar_benign$phyloP100way_vertebrate_pred[i] <- "-"
  } else if (clinvar_benign$phyloP100way_vertebrate[i] > 2) {
    clinvar_benign$phyloP100way_vertebrate_pred[i] <- 1
  } else if (clinvar_benign$phyloP100way_vertebrate[i] < 2) {
    clinvar_benign$phyloP100way_vertebrate_pred[i] <- 0
  }
}
```

Se crea la clasificación final para las variantes benignas.

```{r}
clinvar_benign$classification <- 0
```

El último paso previo a la comparación de las métricas es la unión de ambos datasets, variantes patógenas y benignas, en un único dataframe, y la selección de las columnas correspondientes a las predicciones en sí, obviando el resto de anotaciones del repositorio.

```{r}
clinvar_pathogenic_benign <- rbind(clinvar_pathogenic, clinvar_benign)
predictors <- clinvar_pathogenic_benign[,c("SIFT_pred", "MetaSVM_pred", "MetaLR_pred", "PrimateAI_pred", "DEOGEN2_pred", "BayesDel_addAF_pred", "ClinPred_pred", "LIST.S2_pred", "fathmm.MKL_coding_pred", "VEST4_pred", "REVEL_pred", "CADD_hg19_pred", "DANN_pred", "Eigen_pred", "GERP_pred", "phyloP100way_vertebrate_pred", "classification")]
```

En el siguiente bloque de código se lleva a cabo la obtanción de las diferentes métricas de evaluación de los predictores, en base a los cálculos de las matrices de confusión a partir del paquete caret. Además de esto mediante el paquete ROCit dibujamos las gráficas ROC para cada predictor y calculamos el valor del área bajo la curva, o AUC, una de las métricas indispensables que se suelen utilizar para evaluar el rendimiento global de un método de clasificación. Todos estas métricas y cálculos se van guardando en un dataframe para visualizarlo posteriormente. Las diferentes métricas se describirán en las correspondientes tareas aparte, junto con la explicación de los métodos de predicción y conjuntos de datos escogidos para el análisis.

```{r}
for (i in 1:16) {
  confMat <- confusionMatrix(as.factor(predictors[which(predictors[,i] != "-"), i]), as.factor(predictors[which(predictors[,i] != "-"), ncol(predictors)]), mode = "everything", positive = "1")
  ROC <- rocit(score = as.numeric(predictors[which(predictors[,i] != "-"), i]), class = predictors[which(predictors[,i] != "-"), ncol(predictors)])
  if (i == 1) {
    metrics <- c(confMat$byClass, confMat$overall, mcc(as.factor(predictors[which(predictors[,i] != "-"), i]), as.factor(predictors[which(predictors[,i] != "-"), ncol(predictors)])), ROC$AUC)
    plot(ROC, col = c(i, "gray50"), legend = FALSE, YIndex = FALSE)
  }
  if (i != 1) {
    metrics <- rbind(metrics, c(confMat$byClass, confMat$overall, mcc(as.factor(predictors[which(predictors[,i] != "-"), i]), as.factor(predictors[which(predictors[,i] != "-"), ncol(predictors)])), ROC$AUC))
    lines(ROC$TPR ~ ROC$FPR, col = i, lwd = 2)
  }
}

legend("bottomright", col = seq(1, 16), c("SIFT_pred", "MetaSVM_pred", "MetaLR_pred", "PrimateAI_pred", "DEOGEN2_pred", "BayesDel_addAF_pred", "ClinPred_pred", "LIST.S2_pred", "fathmm.MKL_coding_pred", "VEST4_pred", "REVEL_pred", "CADD_hg19_pred", "DANN_pred", "Eigen_pred", "GERP_pred", "phyloP100way_vertebrate_pred"), lwd = 2, cex = 0.65)
```

Modificamos ligeramente la estructura y la dexcripción de los campos del dataframe construido para una mejor comprensión.

```{r}
```

```
rownames(metrics) <- c("SIFT_pred", "MetaSVM_pred", "MetaLR_pred", "PrimateAI_pred", "DEOGEN2_pred", "BayesDel_addAF_pred", "ClinPred_pred", "LIST.S2_pred", "fathmm.MKL_coding_pred", "VEST4_pred", "REVEL_pred", "CADD_hg19_pred", "DANN_pred", "Eigen_pred", "GERP_pred", "phyloP100way_vertebrate_pred")

metrics <- data.frame(metrics)

names(metrics)[names(metrics) == "V19"] <- "MCC"
names(metrics)[names(metrics) == "V20"] <- "AUC"

metrics <- metrics[, c("Sensitivity", "Specificity", "Neg.Pred.Value", "Precision", "F1", "Prevalence", "Detection.Rate", "Detection.Prevalence", "Balanced.Accuracy", "Accuracy", "Kappa", "MCC", "AUC")]
```

Finalmente, mostramos la tabla completa representando las diferentes métricas calculadas para cada uno de los predictores, con la que podemos analizar rápidamente su rendimiento en base a distintos aspectos y elaborar posteriormente las conclusiones pertinentes. La tabla se encuentra ordenada por el valor AUC de mayor a menor, por lo que teóricamente representaríamos arriba los mejores predictores, siempre en base única y exclusivamente a esta métrica, aunque el análisis se concretará según todas y cada una de las métricas.

```
```{r}
kable(metrics[order(-metrics$AUC),]) %>% kable_styling(bootstrap_options = "striped", full_width = F, font_size = 9.5)
```
```

A continuación se elabora el siguiente bloque de análisis correspondiente a la concordancia entre diferentes predictores, es decir, a la medida de cuánto se parecen las clasificaciones de los diferentes métodos. Esta medida es interesante especialmente porque nos indica si la aplicación de este tipo de métodos en investigaciones clínicas es factible o no, teniendo en cuenta que en muchas ocasiones es necesario que una gran cantidad de predicciones sean concordantes para establecer la patogenicidad de una variante. Se crean en este caso sendos heatmaps para los dos conjuntos de datos de forma separada para visualizar rápidamente cómo varían las predicciones según un método u otro. En color rojo se representa la clasificación patógena, mientras que en azul se representa la predicción benigna.

```
```{r, warning=FALSE}
predictors <- apply(predictors, 2, as.numeric)
predictors_heat_pathogenic <- melt(predictors[which(predictors[,17] == 1),])
predictors_heat_benign <- melt(predictors[which(predictors[,17] == 0),])
p <- ggplot(data = data.frame(predictors_heat_pathogenic), aes(x = X2, y = X1, fill= as.factor(value))) + geom_tile() + scale_fill_manual(values = c("#4393C3", "#F4A582"), labels=c(0, 1))+ scale_y_discrete(expand = c(0, 0))+ scale_x_discrete(expand = c(0, 0))
p + theme(axis.text.y=element_blank()) + theme(axis.ticks.y=element_blank()) + ylab("Variants") + theme(axis.title.y=element_text(size=10)) + xlab("Predictors") + theme(axis.title.x=element_text(size=10)) + theme(axis.text.x=element_text(angle=90,size=10,hjust=1,vjust=.5)) + theme(legend.position="none")

p <- ggplot(data = data.frame(predictors_heat_benign), aes(x = X2, y = X1, fill= as.factor(value))) + geom_tile() + scale_fill_manual(values = c("#4393C3", "#F4A582"), labels=c(0, 1))+ scale_y_discrete(expand = c(0, 0))+ scale_x_discrete(expand = c(0, 0))
p + theme(axis.text.y=element_blank()) + theme(axis.ticks.y=element_blank()) + ylab("Variants") + theme(axis.title.y=element_text(size=10)) + xlab("Predictors") + theme(axis.title.x=element_text(size=10)) + theme(axis.text.x=element_text(angle=90,size=10,hjust=1,vjust=.5)) + theme(legend.position="none")
```
```

Se lleva a cabo entonces el análisis de la concordancia para todos los predictores, calculando cuál es el porcentaje de variantes del conjunto de datos con la que se obtiene la misma predicción para todos ellos, tanto para las variantes patógenas como benignas.

Vemos que obtenemos unos porcentajes del 19'66 y 15'77 %, respectivamente, dando a entender lo complicado que resulta que todo este abanico de predictores de patogenicidad concuerden en su resultado, poniendo en gran valor este trabajo de evaluación y caracterización para determinar cuál o cuáles son los predictores óptimos a utilizar para obtener un conjunto de predicciones precisas.

```
```{r}
predictors_pathogenic <- predictors[which(predictors[,17] == 1),]
predictors_benign <- predictors[which(predictors[,17] == 0),]
concordance_pathogenic <- 0
```
```

```
for (i in 1:nrow(predictors_pathogenic)) {
  if (length(unique(predictors_pathogenic[i,])) == 1) {
    concordance_pathogenic <- concordance_pathogenic + 1
  }
}

concordance_benign <- 0
for (i in 1:nrow(predictors_benign)) {
  if (length(unique(predictors_benign[i,])) == 1) {
    concordance_benign <- concordance_benign + 1
  }
}

print(concordance_pathogenic/nrow(predictors_pathogenic)*100)
print(concordance_benign/nrow(predictors_benign)*100)
```

Por último, mostraremos cuáles son los pares de predictores que concuerdan más en sus predicciones, comparándolos dos a dos mediante el siguiente bloque de código. Aparecen en las tablas los 10 pares de métodos con mejores porcentajes de concordancia, ordenados de mayor a menor.

```{r}
predictors_pathogenic <- data.frame(predictors_pathogenic)
predictors_benign <- data.frame(predictors_benign)

concordance <- data.frame()
cont <- 1
for (i in 1:16) {
  for (j in 1:16) {
    porcentajes <- 0
    if (i != j) {
      for (k in 1:nrow(predictors_pathogenic)) {
        if ((identical(predictors_pathogenic[k,i], predictors_pathogenic[k,j])) == TRUE) {
          porcentajes <- porcentajes + 1
        }
      }
      cont <- cont + 1
      concordance[cont, 1] <- paste(names(data.frame(predictors_pathogenic))[i], "/",
names(data.frame(predictors_pathogenic))[j])
      concordance[cont, 2] <- porcentajes/nrow(predictors_pathogenic)*100
    }
  }
}

concordance <- concordance[!duplicated(concordance$V2),]

kable(head(na.omit(concordance[order(-concordance$V2),]), 10)) %>% kable_styling(bootstrap_options =
"striped", full_width = F, font_size = 9.5)

concordance <- data.frame()
cont <- 1
for (i in 1:16) {
  for (j in 1:16) {
    porcentajes <- 0
    if (i != j) {
      for (k in 1:nrow(predictors_benign)) {
        if ((identical(predictors_benign[k,i], predictors_benign[k,j])) == TRUE) {
          porcentajes <- porcentajes + 1
        }
      }
      cont <- cont + 1
      concordance[cont, 1] <- paste(names(data.frame(predictors_benign))[i], "/",
names(data.frame(predictors_benign))[j])
      concordance[cont, 2] <- porcentajes/nrow(predictors_benign)*100
    }
  }
}
```

concordance <- concordance[!duplicated(concordance$V2),]

kable(head(na.omit(concordance[order(-concordance$V2),]), 10)) %>% kable_styling(bootstrap_options = "striped", full_width = F, font_size = 9.5)
```

*Exportamos la tabla con los predictores en formato CSV.*

```*{r}*
write.table(predictors, file = "predictors_clinvar.csv", sep = ",", quote = FALSE, row.names = FALSE)
```

## 7.2 Código en formato R Markdown de los análisis realizados para la evaluación de predictores de patogenicidad del conjunto de datos de IARC/ICGC

*---*
*title: "Evaluación predictores de patogenicidad con variantes somáticas de TP53 y ICGC"*
*author: "Víctor Manuel Duarte Rute"*
*date: "2/11/2020"*
*output:*
*  html_document:*
*    df_print: paged*
*  pdf_document: default*
*---*

*<style type="text/css">*
*.main-container {*
*  max-width: 1800px;*
*  margin-left: auto;*
*  margin-right: auto;*
*}*
*</style>*

```*{r setup, include=FALSE, warning=FALSE}*
knitr::opts_chunk$set(echo = TRUE)
library(caret)
library(pROC)
library(ROCit)
library(mltools)
library(knitr)
library(kableExtra)
library(ggplot2)
library(reshape)
```

*En este informe basado en R Markdown vamos a ir realizando los diferentes procesamientos de los datos y los análisis pertinentes para evaluar el rendimiento de los predictores de patogenicidad elegidos. Para ello partimos de un fichero tabulado procedente de la anotación de las variantes con el repositorio dbNSFP, que incluye multitud de información interesante para caracterizar todo tipo de variantes, para obtener las predicciones de patogenicidad de las variantes presentes en nuestro dataset. Para maximizar la precisión de los análisis se han llevado a cabo preprocesamientos y filtros previos para quedarnos con aquellas variantes más indicadas para este tipo de algoritmos; más concretamente, nos quedamos con variantes con una consecuencia de 'missense_variant' catalogada por Ensembl, ya que son aquéllas que provocan un cambio de aminoácido y por tanto las que más interesan por su potencial clínico. Asímismo, nos quedamos ocn variantes de tipo SNV, o Single Nucleotide Variation, mucho más comunes y sencillas de analizar por este tipo de predictores in silico.*

```*{r}*
setwd("D:/MÁSTER BIOESTADÍSTICA UOC/TFM/files (new analysis)/RMarkdowns_reports")
```

*Cargamos los datos tabulados en sendas tablas, una para las clasificadas como patógenas y otra para las benignas.*
```

```{r}
pathogenic <- read.csv("somatic_missense_TP53_ICGC_pathogenic_dbNSFP.out", header = TRUE, sep = "\t")
benign <- read.csv("somatic_missense_TP53_ICGC_benign_dbNSFP.out", header = TRUE, sep = "\t")
```

Creamos la primera función encargada de reformatear algunos de los campos de la tabla, concretamente aquellos predictores que poseen múltiples valores debido a la presencia de diferentes transcritos. La aplicación de esta función hace que se haga un merge de todas las clasificaciones para cada variante y nos quedemos con la etiqueta mayoritaria, sea patógena (D, o Deleterious, a partir de ahora etiquetada como '1') o benigna (T, o Tolerated, a partir de ahora etiquetada como '0').

```{r}
reformat_predictions <- function(predictions) {
  predictions_reformat <- c()
  for (i in 1:length(predictions)) {
    split_predictions <- strsplit(predictions[i], ";")
    tolerated <- 0
    deleterious <- 0
    for (j in 1:lengths(split_predictions)) {
      if (split_predictions[[1]][j] == "T") {
        tolerated <- tolerated + 1
        }
      if (split_predictions[[1]][j] == "D") {
        deleterious <- deleterious + 1
        }
    }

    if (deleterious > tolerated)
      predictions_reformat[i] = 1
    if (deleterious < tolerated)
      predictions_reformat[i] = 0
    if (tolerated == 0 && deleterious == 0)
      predictions_reformat[i] = "-"
    if (tolerated == deleterious && tolerated != 0)
      predictions_reformat[i] = 1
  }

  return(predictions_reformat)
}
```

Aplicamos dicha función a los 3 predictores que poseen esta clasificación múltiple, como son SIFT, DEOGEN2 y LIST S2.

```{r}
pathogenic$SIFT_pred <- reformat_predictions(pathogenic$SIFT_pred)
pathogenic$DEOGEN2_pred <- reformat_predictions(pathogenic$DEOGEN2_pred)
pathogenic$LIST.S2_pred <- reformat_predictions(pathogenic$LIST.S2_pred)

benign$SIFT_pred <- reformat_predictions(benign$SIFT_pred)
benign$DEOGEN2_pred <- reformat_predictions(benign$DEOGEN2_pred)
benign$LIST.S2_pred <- reformat_predictions(benign$LIST.S2_pred)
```

La siguiente función que creamos se encarga de reformatear la gran mayoría de los predictores para tener la clasificación numérica que hemos comentado anteriormente, aunque en este caso es más sencillo al no ser valores múltiples.

```{r}
reformat_points <- function(predictions) {
  for (i in 1:length(predictions)) {
    if (predictions[i] == ".") {
      predictions[i] <- "-"
    }
    if (predictions[i] == "D") {
      predictions[i] <- 1
```

```
    }
    if (predictions[i] == "T" || predictions[i] == "N") {
      predictions[i] <- 0
    }
  }

  return(predictions)
}
```

Aplicamos la función anterior con los predictores MetaLR, MetaSVM, PrimateAI, fathmm_MKL, BayesDel y ClinPred.

```{r}
pathogenic$MetaLR_pred <- reformat_points(pathogenic$MetaLR_pred)
pathogenic$MetaSVM_pred <- reformat_points(pathogenic$MetaSVM_pred)
pathogenic$PrimateAI_pred <- reformat_points(pathogenic$PrimateAI_pred)
pathogenic$fathmm.MKL_coding_pred <- reformat_points(pathogenic$fathmm.MKL_coding_pred)
pathogenic$BayesDel_addAF_pred <- reformat_points(pathogenic$BayesDel_addAF_pred)
pathogenic$ClinPred_pred <- reformat_points(pathogenic$ClinPred_pred)

benign$MetaLR_pred <- reformat_points(benign$MetaLR_pred)
benign$MetaSVM_pred <- reformat_points(benign$MetaSVM_pred)
benign$PrimateAI_pred <- reformat_points(benign$PrimateAI_pred)
benign$fathmm.MKL_coding_pred <- reformat_points(benign$fathmm.MKL_coding_pred)
benign$BayesDel_addAF_pred <- reformat_points(benign$BayesDel_addAF_pred)
benign$ClinPred_pred <- reformat_points(benign$ClinPred_pred)
```

A continuación los siguientes bloques de código se encargan de reformatear el resto de predictores a analizar, cuya peculiaridad es que no poseen de entrada una clasificación categórica, por lo que es necesario que nosotros mismos convirtamos la predicción numérica (cada predictor con su propio rango de valores) en una clasificación basada en etiqueta. Esta conversión se realiza en base a las propias estimaciones de los desarrolladores de los predictores, que proponen un cutoff o valor determinado a partir del cual establecer la división. Cada bloque por tanto se dirige a la transformación de un predictor distinto según su propio rango de valores.

VEST4

```{r}
for (i in 1:length(pathogenic$VEST4_score)) {
  split_scores <- strsplit(pathogenic$VEST4_score[i], ";")
  highest_score <- "NA"
  boolean <- 0
  for (j in 1:lengths(split_scores)) {
    if (grepl("^[0-9]", split_scores[[1]][j])) {
      score <- as.numeric(split_scores[[1]][j])
      if (boolean == 0) {
        highest_score <- score
        boolean <- 1
      }
      if (score > highest_score) {
        highest_score <- score
      }
    }
  }
  if (highest_score == "NA") {
    pathogenic$VEST4_pred[i] <- "-"
  } else if (highest_score >= 0.5) {
    pathogenic$VEST4_pred[i] <- 1
  } else if (highest_score < 0.5) {
    pathogenic$VEST4_pred[i] <- 0
  }
}
```

REVEL

```{r}
for (i in 1:length(pathogenic$REVEL_score)) {
  split_scores <- strsplit(pathogenic$REVEL_score[i], ";")
  highest_score <- "NA"
  boolean <- 0
  for (j in 1:lengths(split_scores)) {
    if (grepl("^[0-9]", split_scores[[1]][j])) {
      score <- as.numeric(split_scores[[1]][j])
      if (boolean == 0) {
        highest_score <- score
        boolean <- 1
      }
      if (score > highest_score) {
        highest_score <- score
      }
    }
  }
  if (highest_score == "NA") {
    pathogenic$REVEL_pred[i] <- "-"
  } else if (highest_score >= 0.4) {
    pathogenic$REVEL_pred[i] <- 1
  } else if (highest_score < 0.4) {
    pathogenic$REVEL_pred[i] <- 0
  }
}
```

CADD

```{r}
for (i in 1:length(pathogenic$CADD_phred_hg19)) {
  if (pathogenic$CADD_phred_hg19[i] == ".") {
    pathogenic$CADD_hg19_pred[i] <- "-"
  } else if (pathogenic$CADD_phred_hg19[i] > 20) {
    pathogenic$CADD_hg19_pred[i] <- 1
  } else if (pathogenic$CADD_phred_hg19[i] <= 20) {
    pathogenic$CADD_hg19_pred[i] <- 0
  }
}
```

DANN

```{r}
for (i in 1:length(pathogenic$DANN_score)) {
  if (pathogenic$DANN_score[i] == ".") {
    pathogenic$DANN_pred[i] <- "-"
  } else if (pathogenic$DANN_score[i] >= 0.99) {
    pathogenic$DANN_pred[i] <- 1
  } else if (pathogenic$DANN_score[i] < 0.99) {
    pathogenic$DANN_pred[i] <- 0
  }
}
```

Eigen

```{r}
for (i in 1:length(pathogenic$Eigen.raw_coding)) {
  if (pathogenic$Eigen.raw_coding[i] == ".") {
    pathogenic$Eigen_pred[i] <- "-"
  } else if (pathogenic$Eigen.raw_coding[i] >= 0) {
    pathogenic$Eigen_pred[i] <- 1
  } else if (pathogenic$Eigen.raw_coding[i] < 0) {
    pathogenic$Eigen_pred[i] <- 0
  }
```

```
}
```

GERP

```{r}
for (i in 1:length(pathogenic$GERP.._RS)) {
  if (pathogenic$GERP.._RS[i] == ".") {
    pathogenic$GERP_pred[i] <- "-"
  } else if (pathogenic$GERP.._RS[i] >= 2) {
    pathogenic$GERP_pred[i] <- 1
  } else if (pathogenic$GERP.._RS[i] < 2) {
    pathogenic$GERP_pred[i] <- 0
  }
}
```

PhyloP

```{r}
for (i in 1:length(pathogenic$phyloP100way_vertebrate)) {
  if (pathogenic$phyloP100way_vertebrate[i] == ".") {
    pathogenic$phyloP100way_vertebrate_pred[i] <- "-"
  } else if (pathogenic$phyloP100way_vertebrate[i] > 2) {
    pathogenic$phyloP100way_vertebrate_pred[i] <- 1
  } else if (pathogenic$phyloP100way_vertebrate[i] < 2) {
    pathogenic$phyloP100way_vertebrate_pred[i] <- 0
  }
}
```

Finalmente creamos la columna de clasificación, en la que se especifica el valor real de la patogenicidad de la variante según las características del conjunto de datos.

```{r}
pathogenic$classification <- 1
```

A continuación se lleva a cabo la misma transformación anterior con el conjunto de datos de variantes benignas.

VEST4

```{r}
for (i in 1:length(benign$VEST4_score)) {
  split_scores <- strsplit(benign$VEST4_score[i], ";")
  highest_score <- "NA"
  boolean <- 0
  for (j in 1:lengths(split_scores)) {
    if (grepl("^[0-9]", split_scores[[1]][j])) {
      score <- as.numeric(split_scores[[1]][j])
      if (boolean == 0) {
        highest_score <- score
        boolean <- 1
      }
      if (score > highest_score) {
        highest_score <- score
      }
    }
  }
  if (highest_score == "NA") {
    benign$VEST4_pred[i] <- "-"
  } else if (highest_score >= 0.5) {
    benign$VEST4_pred[i] <- 1
  } else if (highest_score < 0.5) {
    benign$VEST4_pred[i] <- 0
  }
```

```
}
```

REVEL

```{r}
for (i in 1:length(benign$REVEL_score)) {
  split_scores <- strsplit(benign$REVEL_score[i], ";")
  highest_score <- "NA"
  boolean <- 0
  for (j in 1:lengths(split_scores)) {
    if (grepl("^[0-9]", split_scores[[1]][j])) {
      score <- as.numeric(split_scores[[1]][j])
      if (boolean == 0) {
        highest_score <- score
        boolean <- 1
      }
      if (score > highest_score) {
        highest_score <- score
      }
    }
  }
  if (highest_score == "NA") {
    benign$REVEL_pred[i] <- "-"
  } else if (highest_score >= 0.4) {
    benign$REVEL_pred[i] <- 1
  } else if (highest_score < 0.4) {
    benign$REVEL_pred[i] <- 0
  }
}
```

CADD

```{r}
for (i in 1:length(benign$CADD_phred_hg19)) {
  if (benign$CADD_phred_hg19[i] == ".") {
    benign$CADD_hg19_pred[i] <- "-"
  } else if (benign$CADD_phred_hg19[i] > 20) {
    benign$CADD_hg19_pred[i] <- 1
  } else if (benign$CADD_phred_hg19[i] <= 20) {
    benign$CADD_hg19_pred[i] <- 0
  }
}
```

DANN

```{r}
for (i in 1:length(benign$DANN_score)) {
  if (benign$DANN_score[i] == ".") {
    benign$DANN_pred[i] <- "-"
  } else if (benign$DANN_score[i] >= 0.99) {
    benign$DANN_pred[i] <- 1
  } else if (benign$DANN_score[i] < 0.99) {
    benign$DANN_pred[i] <- 0
  }
}
```

Eigen

```{r}
for (i in 1:length(benign$Eigen.raw_coding)) {
  if (benign$Eigen.raw_coding[i] == ".") {
    benign$Eigen_pred[i] <- "-"
  } else if (benign$Eigen.raw_coding[i] >= 0) {
```

```
    benign$Eigen_pred[i] <- 1
  } else if (benign$Eigen.raw_coding[i] < 0) {
    benign$Eigen_pred[i] <- 0
  }
}
```

GERP

```{r}
for (i in 1:length(benign$GERP.._RS)) {
  if (benign$GERP.._RS[i] == ".") {
    benign$GERP_pred[i] <- "-"
  } else if (benign$GERP.._RS[i] >= 2) {
    benign$GERP_pred[i] <- 1
  } else if (benign$GERP.._RS[i] < 2) {
    benign$GERP_pred[i] <- 0
  }
}
```

PhyloP

```{r}
for (i in 1:length(benign$phyloP100way_vertebrate)) {
  if (benign$phyloP100way_vertebrate[i] == ".") {
    benign$phyloP100way_vertebrate_pred[i] <- "-"
  } else if (benign$phyloP100way_vertebrate[i] > 2) {
    benign$phyloP100way_vertebrate_pred[i] <- 1
  } else if (benign$phyloP100way_vertebrate[i] < 2) {
    benign$phyloP100way_vertebrate_pred[i] <- 0
  }
}
```

Se crea la clasificación final para las variantes benignas.

```{r}
benign$classification <- 0
```

El último paso previo a la comparación de las métricas es la unión de ambos datasets, variantes patógenas y benignas, en un único dataframe, y la selección de las columnas correspondientes a las predicciones en sí, obviando el resto de anotaciones del repositorio.

```{r}
pathogenic_benign <- rbind(pathogenic, benign)
predictors <- pathogenic_benign[,c("SIFT_pred", "MetaSVM_pred", "MetaLR_pred", "PrimateAI_pred", "DEOGEN2_pred", "BayesDel_addAF_pred", "ClinPred_pred", "LIST.S2_pred", "fathmm.MKL_coding_pred", "VEST4_pred", "REVEL_pred", "CADD_hg19_pred", "DANN_pred", "Eigen_pred", "GERP_pred", "phyloP100way_vertebrate_pred", "classification")]
```

En el siguiente bloque de código se lleva a cabo la obtanción de las diferentes métricas de evaluación de los predictores, en base a los cálculos de las matrices de confusión a partir del paquete caret. Además de esto mediante el paquete ROCit dibujamos las gráficas ROC para cada predictor y calculamos el valor del área bajo la curva, o AUC, una de las métricas indispensables que se suelen utilizar para evaluar el rendimiento global de un método de clasificación. Todos estas métricas y cálculos se van guardando en un dataframe para visualizarlo posteriormente. Las diferentes métricas se describirán en las correspondientes tareas aparte, junto con la explicación de los métodos de predicción y conjuntos de datos escogidos para el análisis.

```{r}
for (i in 1:16) {
  confMat <- confusionMatrix(as.factor(predictors[which(predictors[,i] != "-"), i]), as.factor(predictors[which(predictors[,i] != "-"), ncol(predictors)]), mode = "everything", positive = "1")
```

```
 ROC <- rocit(score = as.numeric(predictors[which(predictors[,i] != "-"), i]), class = predictors[which(predictors[,i] != "-"), ncol(predictors)])
  if (i == 1) {
    metrics <- c(confMat$byClass, confMat$overall, mcc(as.factor(predictors[which(predictors[,i] != "-"), i]), as.factor(predictors[which(predictors[,i] != "-"), ncol(predictors)])), ROC$AUC)
    plot(ROC, col = c(i, "gray50"), legend = FALSE, YIndex = FALSE)
  }
  if (i != 1) {
    metrics <- rbind(metrics, c(confMat$byClass, confMat$overall, mcc(as.factor(predictors[which(predictors[,i] != "-"), i]), as.factor(predictors[which(predictors[,i] != "-"), ncol(predictors)])), ROC$AUC))
    lines(ROC$TPR ~ ROC$FPR, col = i, lwd = 2)
  }
}

legend("bottomright", col = seq(1, 16), c("SIFT_pred", "MetaSVM_pred", "MetaLR_pred", "PrimateAI_pred", "DEOGEN2_pred", "BayesDel_addAF_pred", "ClinPred_pred", "LIST.S2_pred", "fathmm.MKL_coding_pred", "VEST4_pred", "REVEL_pred", "CADD_hg19_pred", "DANN_pred", "Eigen_pred", "GERP_pred", "phyloP100way_vertebrate_pred"), lwd = 2, cex = 0.65)
```

Modificamos ligeramente la estructura y la dexcripción de los campos del dataframe construido para una mejor comprensión.

```{r}
rownames(metrics) <- c("SIFT_pred", "MetaSVM_pred", "MetaLR_pred", "PrimateAI_pred", "DEOGEN2_pred", "BayesDel_addAF_pred", "ClinPred_pred", "LIST.S2_pred", "fathmm.MKL_coding_pred", "VEST4_pred", "REVEL_pred", "CADD_hg19_pred", "DANN_pred", "Eigen_pred", "GERP_pred", "phyloP100way_vertebrate_pred")

metrics <- data.frame(metrics)

names(metrics)[names(metrics) == "V19"] <- "MCC"
names(metrics)[names(metrics) == "V20"] <- "AUC"

metrics <- metrics[, c("Sensitivity", "Specificity", "Neg.Pred.Value", "Precision", "F1", "Prevalence", "Detection.Rate", "Detection.Prevalence", "Balanced.Accuracy", "Accuracy", "Kappa", "MCC", "AUC")]
```

Finalmente, mostramos la tabla completa representando las diferentes métricas calculadas para cada uno de los predictores, con la que podemos analizar rápidamente su rendimiento en base a distintos aspectos y elaborar posteriormente las conclusiones pertinentes. La tabla se encuentra ordenada por el valor AUC de mayor a menor, por lo que teóricamente representaríamos arriba los mejores predictores, siempre en base única y exclusivamente a esta métrica, aunque el análisis se concretará según todas y cada una de las métricas.

```{r}
kable(metrics[order(-metrics$AUC),]) %>% kable_styling(bootstrap_options = "striped", full_width = F, font_size = 9.5)
```

A continuación se elabora el siguiente bloque de análisis correspondiente a la concordancia entre diferentes predictores, es decir, a la medida de cuánto se parecen las clasificaciones de los diferentes métodos. Esta medida es interesante especialmente porque nos indica si la aplicación de este tipo de métodos en investigaciones clínicas es factible o no, teniendo en cuenta que en muchas ocasiones es necesario que una gran cantidad de predicciones sean concordantes para establecer la patogenicidad de una variante. Se crean en este caso sendos heatmaps para los dos conjuntos de datos de forma separada para visualizar rápidamente cómo varían las predicciones según un método u otro. En color rojo se representa la clasificación patógena, mientras que en azul se representa la predicción benigna.

```{r, warning=FALSE}
predictors <- apply(predictors, 2, as.numeric)
predictors_heat_pathogenic <- melt(predictors[which(predictors[,17] == 1),])
predictors_heat_benign <- melt(predictors[which(predictors[,17] == 0),])
p <- ggplot(data = data.frame(predictors_heat_pathogenic), aes(x = X2, y = X1, fill= as.factor(value))) + geom_tile() + scale_fill_manual(values = c("#4393C3", "#F4A582"), labels=c(0, 1))+ scale_y_discrete(expand = c(0, 0))+ scale_x_discrete(expand = c(0, 0))
```

```
p + theme(axis.text.y=element_blank()) + theme(axis.ticks.y=element_blank()) + ylab("Variants") + theme(axis.title.y=element_text(size=10)) + xlab("Predictors") + theme(axis.title.x=element_text(size=10)) + theme(axis.text.x=element_text(angle=90,size=10,hjust=1,vjust=.5)) + theme(legend.position="none")

p <- ggplot(data = data.frame(predictors_heat_benign), aes(x = X2, y = X1, fill= as.factor(value))) + geom_tile() + scale_fill_manual(values = c("#4393C3", "#F4A582"), labels=c(0, 1))+ scale_y_discrete(expand = c(0, 0))+ scale_x_discrete(expand = c(0, 0))
p + theme(axis.text.y=element_blank()) + theme(axis.ticks.y=element_blank()) + ylab("Variants") + theme(axis.title.y=element_text(size=10)) + xlab("Predictors") + theme(axis.title.x=element_text(size=10)) + theme(axis.text.x=element_text(angle=90,size=10,hjust=1,vjust=.5)) + theme(legend.position="none")
```

Se lleva a cabo entonces el análisis de la concordancia para todos los predictores, calculando cuál es el porcentaje de variantes del conjunto de datos con la que se obtiene la misma predicción para todos ellos, tanto para las variantes patógenas como benignas.

Vemos que obtenemos unos porcentajes del 13'77 y 0'51 %, respectivamente, dando a entender lo complicado que resulta que todo este abanico de predictores de patogenicidad concuerden en su resultado, poniendo en gran valor este trabajo de evaluación y caracterización para determinar cuál o cuáles son los predictores óptimos a utilizar para obtener un conjunto de predicciones precisas.

```{r}
predictors_pathogenic <- predictors[which(predictors[,17] == 1),]
predictors_benign <- predictors[which(predictors[,17] == 0),]
concordance_pathogenic <- 0
for (i in 1:nrow(predictors_pathogenic)) {
  if (length(unique(predictors_pathogenic[i,])) == 1) {
    concordance_pathogenic <- concordance_pathogenic + 1
  }
}

concordance_benign <- 0
for (i in 1:nrow(predictors_benign)) {
  if (length(unique(predictors_benign[i,])) == 1) {
    concordance_benign <- concordance_benign + 1
  }
}

print(concordance_pathogenic/nrow(predictors_pathogenic)*100)
print(concordance_benign/nrow(predictors_benign)*100)
```

Por último, mostraremos cuáles son los pares de predictores que concuerdan más en sus predicciones, comparándolos dos a dos mediante el siguiente bloque de código. Aparecen en las tablas los 10 pares de métodos con mejores porcentajes de concordancia, ordenados de mayor a menor.

```{r}
predictors_pathogenic <- data.frame(predictors_pathogenic)
predictors_benign <- data.frame(predictors_benign)

concordance <- data.frame()
cont <- 1
for (i in 1:16) {
  for (j in 1:16) {
    porcentajes <- 0
    if (i != j) {
      for (k in 1:nrow(predictors_pathogenic)) {
        if ((identical(predictors_pathogenic[k,i], predictors_pathogenic[k,j])) == TRUE) {
          porcentajes <- porcentajes + 1
        }
      }
      cont <- cont + 1
      concordance[cont, 1] <- paste(names(data.frame(predictors_pathogenic))[i], "/", names(data.frame(predictors_pathogenic))[j])
      concordance[cont, 2] <- porcentajes/nrow(predictors_pathogenic)*100
    }
  }
```

```
}

concordance <- concordance[!duplicated(concordance$V2),]

kable(head(na.omit(concordance[order(-concordance$V2),]), 10)) %>% kable_styling(bootstrap_options = "striped", full_width = F, font_size = 9.5)

concordance <- data.frame()
cont <- 1
for (i in 1:16) {
  for (j in 1:16) {
    porcentajes <- 0
    if (i != j) {
      for (k in 1:nrow(predictors_benign)) {
        if ((identical(predictors_benign[k,i], predictors_benign[k,j])) == TRUE) {
          porcentajes <- porcentajes + 1
        }
      }
      cont <- cont + 1
      concordance[cont, 1] <- paste(names(data.frame(predictors_benign))[i], "/", names(data.frame(predictors_benign))[j])
      concordance[cont, 2] <- porcentajes/nrow(predictors_benign)*100
    }
  }
}

concordance <- concordance[!duplicated(concordance$V2),]

kable(head(na.omit(concordance[order(-concordance$V2),]), 10)) %>% kable_styling(bootstrap_options = "striped", full_width = F, font_size = 9.5)
```

Exportamos la tabla con los predictores en formato CSV.

```{r}
write.table(predictors, file = "predictors_somatic.csv", sep = ",", quote = FALSE, row.names = FALSE)
```

7.3 Código en formato R Markdown de los análisis realizados para la evaluación de predictores de patogenicidad del conjunto de datos de GIAB

```
---
title: "Evaluación predictores de patogenicidad con variantes de la muestra gold standard NA12878"
author: "Víctor Manuel Duarte Rute"
date: "2/11/2020"
output:
  html_document:
    df_print: paged
---

<style type="text/css">
.main-container {
  max-width: 1800px;
  margin-left: auto;
  margin-right: auto;
}
</style>

```{r setup, include=FALSE, warning=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(caret)
library(pROC)
library(ROCit)
library(mltools)
library(knitr)
library(kableExtra)
```
```

library(ggplot2)
library(reshape)
```

*En este informe basado en R Markdown vamos a ir realizando los diferentes procesamientos de los datos y los análisis pertinentes para evaluar el rendimiento de los predictores de patogenicidad elegidos. Para ello partimos de un fichero tabulado procedente de la anotación de las variantes con el repositorio dbNSFP, que incluye multitud de información interesante para caracterizar todo tipo de variantes, para obtener las predicciones de patogenicidad de las variantes presentes en nuestro dataset. Para maximizar la precisión de los análisis se han llevado a cabo preprocesamientos y filtros previos para quedarnos con aquellas variantes más indicadas para este tipo de algoritmos; más concretamente, nos quedamos con variantes con una consecuencia de 'missense_variant' catalogada por Ensembl, ya que son aquéllas que provocan un cambio de aminoácido y por tanto las que más interesan por su potencial clínico. Asímismo, nos quedamos ocn variantes de tipo SNV, o Single Nucleotide Variation, mucho más comunes y sencillas de analizar por este tipo de predictores in silico.*

*A diferencia de los conjuntos de datos de ClinVar y TP53/ICGC las variantes de este análisis no poseen ningún tipo de información clínica con la que comparar las predicciones y poder evaluar las diferentes métricas de clasificación; en su lugar, la obtención de las predicciones para este conjunto de variantes de la muestra gold standard NA12878, pertenecientes a un exoma completo, nos sirve para medir mediante el análisis de concordancia cómo sería la aplicación de este tipo de métodos in silico en un caso real, similar a los que pueden encontrarse en cualquier laboratorio de análisis de datos genéticos con fines clínicos.*

```{r}
setwd("D:/MÁSTER BIOESTADÍSTICA UOC/TFM/files (new analysis)/RMarkdowns_reports")
```

Cargamos los datos tabulados en un único dataframe.

```{r}
dbNSFP <- read.csv("Garvan_NA12878_exome_vep_missense_dbNSFP.out", header = TRUE, sep = "\t")
```

Creamos la primera función encargada de reformatear algunos de los campos de la tabla, concretamente aquellos predictores que poseen múltiples valores debido a la presencia de diferentes transcritos. La aplicación de esta función hace que se haga un merge de todas las clasificaciones para cada variante y nos quedemos con la etiqueta mayoritaria, sea patógena (D, o Deleterious, a partir de ahora etiquetada como '1') o benigna (T, o Tolerated, a partir de ahora etiquetada como '0').

```{r}
reformat_predictions <- function(predictions) {
  predictions_reformat <- c()
  for (i in 1:length(predictions)) {
    split_predictions <- strsplit(predictions[i], ";")
    tolerated <- 0
    deleterious <- 0
    for (j in 1:lengths(split_predictions)) {
      if (split_predictions[[1]][j] == "T") {
        tolerated <- tolerated + 1
        }
      if (split_predictions[[1]][j] == "D") {
        deleterious <- deleterious + 1
        }
    }

    if (deleterious > tolerated)
      predictions_reformat[i] = 1
    if (deleterious < tolerated)
      predictions_reformat[i] = 0
    if (tolerated == 0 && deleterious == 0)
      predictions_reformat[i] = "-"
    if (tolerated == deleterious && tolerated != 0)
      predictions_reformat[i] = 1
  }

  return(predictions_reformat)
```
```

```
}
```

Aplicamos dicha función a los 3 predictores que poseen esta clasificación múltiple, como son SIFT, DEOGEN2 y LIST S2.

```{r}
dbNSFP$SIFT_pred <- reformat_predictions(dbNSFP$SIFT_pred)
dbNSFP$DEOGEN2_pred <- reformat_predictions(dbNSFP$DEOGEN2_pred)
dbNSFP$LIST.S2_pred <- reformat_predictions(dbNSFP$LIST.S2_pred)
```

La siguiente función que creamos se encarga de reformatear la gran mayoría de los predictores para tener la clasificación numérica que hemos comentado anteriormente, aunque en este caso es más sencillo al no ser valores múltiples.

```{r}
reformat_points <- function(predictions) {
  for (i in 1:length(predictions)) {
    if (predictions[i] == ".") {
      predictions[i] <- "-"
    }
    if (predictions[i] == "D") {
      predictions[i] <- 1
    }
    if (predictions[i] == "T" || predictions[i] == "N") {
      predictions[i] <- 0
    }
  }

  return(predictions)
}
```

Aplicamos la función anterior con los predictores MetaLR, MetaSVM, PrimateAI, fathmm_MKL, BayesDel y ClinPred.

```{r}
dbNSFP$MetaLR_pred <- reformat_points(dbNSFP$MetaLR_pred)
dbNSFP$MetaSVM_pred <- reformat_points(dbNSFP$MetaSVM_pred)
dbNSFP$PrimateAI_pred <- reformat_points(dbNSFP$PrimateAI_pred)
dbNSFP$fathmm.MKL_coding_pred <- reformat_points(dbNSFP$fathmm.MKL_coding_pred)
dbNSFP$BayesDel_addAF_pred <- reformat_points(dbNSFP$BayesDel_addAF_pred)
dbNSFP$ClinPred_pred <- reformat_points(dbNSFP$ClinPred_pred)
```

A continuación los siguientes bloques de código se encargan de reformatear el resto de predictores a analizar, cuya peculiaridad es que no poseen de entrada una clasificación categórica, por lo que es necesario que nosotros mismos convirtamos la predicción numérica (cada predictor con su propio rango de valores) en una clasificación basada en etiqueta. Esta conversión se realiza en base a las propias estimaciones de los desarrolladores de los predictores, que proponen un cutoff o valor determinado a partir del cual establecer la división. Cada bloque por tanto se dirige a la transformación de un predictor distinto según su propio rango de valores.

VEST4

```{r}
for (i in 1:length(dbNSFP$VEST4_score)) {
  split_scores <- strsplit(dbNSFP$VEST4_score[i], ";")
  highest_score <- "NA"
  boolean <- 0
  for (j in 1:lengths(split_scores)) {
    if (grepl("^[0-9]", split_scores[[1]][j])) {
      score <- as.numeric(split_scores[[1]][j])
      if (boolean == 0) {
        highest_score <- score
        boolean <- 1
```

```
      }
      if (score > highest_score) {
        highest_score <- score
      }
    }
  }
  if (highest_score == "NA") {
    dbNSFP$VEST4_pred[i] <- "-"
  } else if (highest_score >= 0.5) {
    dbNSFP$VEST4_pred[i] <- 1
  } else if (highest_score < 0.5) {
    dbNSFP$VEST4_pred[i] <- 0
  }
}
```

REVEL

```{r}
for (i in 1:length(dbNSFP$REVEL_score)) {
  split_scores <- strsplit(dbNSFP$REVEL_score[i], ";")
  highest_score <- "NA"
  boolean <- 0
  for (j in 1:lengths(split_scores)) {
    if (grepl("^[0-9]", split_scores[[1]][j])) {
      score <- as.numeric(split_scores[[1]][j])
      if (boolean == 0) {
        highest_score <- score
        boolean <- 1
      }
      if (score > highest_score) {
        highest_score <- score
      }
    }
  }
  if (highest_score == "NA") {
    dbNSFP$REVEL_pred[i] <- "-"
  } else if (highest_score >= 0.4) {
    dbNSFP$REVEL_pred[i] <- 1
  } else if (highest_score < 0.4) {
    dbNSFP$REVEL_pred[i] <- 0
  }
}
```

CADD

```{r}
for (i in 1:length(dbNSFP$CADD_phred_hg19)) {
  if (dbNSFP$CADD_phred_hg19[i] == ".") {
    dbNSFP$CADD_hg19_pred[i] <- "-"
  } else if (dbNSFP$CADD_phred_hg19[i] > 20) {
    dbNSFP$CADD_hg19_pred[i] <- 1
  } else if (dbNSFP$CADD_phred_hg19[i] <= 20) {
    dbNSFP$CADD_hg19_pred[i] <- 0
  }
}
```

DANN

```{r}
for (i in 1:length(dbNSFP$DANN_score)) {
  if (dbNSFP$DANN_score[i] == ".") {
    dbNSFP$DANN_pred[i] <- "-"
  } else if (dbNSFP$DANN_score[i] >= 0.99) {
    dbNSFP$DANN_pred[i] <- 1
```

```
  } else if (dbNSFP$DANN_score[i] < 0.99) {
    dbNSFP$DANN_pred[i] <- 0
  }
}
```

Eigen

```{r}
for (i in 1:length(dbNSFP$Eigen.raw_coding)) {
  if (dbNSFP$Eigen.raw_coding[i] == ".") {
    dbNSFP$Eigen_pred[i] <- "-"
  } else if (dbNSFP$Eigen.raw_coding[i] >= 0) {
    dbNSFP$Eigen_pred[i] <- 1
  } else if (dbNSFP$Eigen.raw_coding[i] < 0) {
    dbNSFP$Eigen_pred[i] <- 0
  }
}
```

GERP

```{r}
for (i in 1:length(dbNSFP$GERP.._RS)) {
  if (dbNSFP$GERP.._RS[i] == ".") {
    dbNSFP$GERP_pred[i] <- "-"
  } else if (dbNSFP$GERP.._RS[i] >= 2) {
    dbNSFP$GERP_pred[i] <- 1
  } else if (dbNSFP$GERP.._RS[i] < 2) {
    dbNSFP$GERP_pred[i] <- 0
  }
}
```

PhyloP

```{r}
for (i in 1:length(dbNSFP$phyloP100way_vertebrate)) {
  if (dbNSFP$phyloP100way_vertebrate[i] == ".") {
    dbNSFP$phyloP100way_vertebrate_pred[i] <- "-"
  } else if (dbNSFP$phyloP100way_vertebrate[i] > 2) {
    dbNSFP$phyloP100way_vertebrate_pred[i] <- 1
  } else if (dbNSFP$phyloP100way_vertebrate[i] < 2) {
    dbNSFP$phyloP100way_vertebrate_pred[i] <- 0
  }
}
```

Seleccionamos solamente los campos relativos a las predicciones, dejando de lado el resto de anotaciones.

```{r}
predictors <- dbNSFP[,c("SIFT_pred", "MetaSVM_pred", "MetaLR_pred", "PrimateAI_pred", "DEOGEN2_pred", "BayesDel_addAF_pred", "ClinPred_pred", "LIST.S2_pred", "fathmm.MKL_coding_pred", "VEST4_pred", "REVEL_pred", "CADD_hg19_pred", "DANN_pred", "Eigen_pred", "GERP_pred", "phyloP100way_vertebrate_pred")]
```

A continuación se elabora el siguiente bloque de análisis correspondiente a la concordancia entre diferentes predictores, es decir, a la medida de cuánto se parecen las clasificaciones de los diferentes métodos. Esta medida es interesante especialmente porque nos indica si la aplicación de este tipo de métodos en investigaciones clínicas es factible o no, teniendo en cuenta que en muchas ocasiones es necesario que una gran cantidad de predicciones sean concordantes para establecer la patogenicidad de una variante. Se crea en este caso un heatmap para visualizar rápidamente cómo varían las predicciones según un método u otro. En color rojo se representa la clasificación patógena, mientras que en azul se representa la predicción benigna.

```{r, warning=FALSE}
predictors <- apply(predictors, 2, as.numeric)
predictors_heat_dbNSFP <- melt(predictors)
p <- ggplot(data = data.frame(predictors_heat_dbNSFP), aes(x = X2, y = X1, fill= as.factor(value))) + geom_tile() + scale_fill_manual(values = c("#4393C3", "#F4A582"), labels=c(0, 1))+ scale_y_discrete(expand = c(0, 0))+ scale_x_discrete(expand = c(0, 0))
p + theme(axis.text.y=element_blank()) + theme(axis.ticks.y=element_blank()) + ylab("Variants") + theme(axis.title.y=element_text(size=10)) + xlab("Predictors") + theme(axis.title.x=element_text(size=10)) + theme(axis.text.x=element_text(angle=90,size=10,hjust=1,vjust=.5)) + theme(legend.position="none")
```

Se lleva a cabo entonces el análisis de la concordancia para todos los predictores, calculando cuál es el porcentaje de variantes del conjunto de datos con la que se obtiene la misma predicción para todos ellos.

Vemos que obtenemos un valor del 30'67 %, dando a entender lo complicado que resulta que todo este abanico de predictores de patogenicidad concuerden en su resultado, poniendo en gran valor este trabajo de evaluación y caracterización para determinar cuál o cuáles son los predictores óptimos a utilizar para obtener un conjunto de predicciones precisas.

Para este caso concreto de análisis, trabajando con un caso real, vemos que el valor de concordancia obtenido es significativamente superior respecto a los dos casos de anñalisis anteriores, indicando que cuando se trabaja con variantes más comunes en la población, o que son relativamente corrientes en la población, las predicciones se ajustan mejor y se obtienen mejores resultados. Esto refuerza la idea de que es posible utilizar predictores in silico para la clasificación de variantes en experimentos de secuenciación masiva, siempre que ajustemos bien los métodos a usar en función del enfoque, el alcande o el tipo de resultados que queramos obtener (no será lo mismo la confirmación de un resultado benigno o negativo con el cribado inicial de variantes para buscar resultados positivos, o potencialmente patógenos).

```{r}
concordance_dbNSFP <- 0
for (i in 1:nrow(predictors)) {
  if (length(unique(predictors[i,])) == 1) {
    concordance_dbNSFP <- concordance_dbNSFP + 1
  }
}

print(concordance_dbNSFP/nrow(predictors)*100)
```

Por último, mostraremos cuáles son los pares de predictores que concuerdan más en sus predicciones, comparándolos dos a dos mediante el siguiente bloque de código. Aparecen en la tabla los 10 pares de métodos con mejores porcentajes de concordancia, ordenados de mayor a menor.

```{r}
predictors <- data.frame(predictors)

concordance <- data.frame()
cont <- 1
for (i in 1:16) {
  for (j in 1:16) {
    porcentajes <- 0
    if (i != j) {
      for (k in 1:nrow(predictors)) {
        if ((identical(predictors[k,i], predictors[k,j])) == TRUE) {
          porcentajes <- porcentajes + 1
        }
      }
      cont <- cont + 1
      concordance[cont, 1] <- paste(names(data.frame(predictors))[i], "/", names(data.frame(predictors))[j])
      concordance[cont, 2] <- porcentajes/nrow(predictors)*100
    }
  }
}

concordance <- concordance[!duplicated(concordance$V2),]
```

kable(head(na.omit(concordance[order(-concordance$V2),]), 10)) %>% kable_styling(bootstrap_options = "striped", full_width = F, font_size = 9.5)
```

*Exportamos la tabla con los predictores en formato CSV.*

```*{r}*
write.table(predictors, file = "predictors_NA12878.csv", sep = ",", quote = FALSE, row.names = FALSE)
```
```