Tesis de Maestría en Bioinformática

Clasificación de variantes en el genoma humano mediante aprendizaje automático y redes de expertos, con énfasis en enfermedades raras: una primera aproximación

---

Estudiante:
Camila Simoes Amaro

Supervisores:
Hugo Naya, Juan Cardelino, Víctor Raggio

Tesis presentada en cumplimiento parcial de los requerimientos para la obtención del título de Magister en Bioinformática

PEDECIBA - Universidad de la República

27 de abril de 2023

Agradecimientos

A las instituciones que permitieron el desarrollo de este trabajo y mi formación. Muchas gracias a la Universidad de la República, a PEDECIBA, y al Institut Pasteur de Montevideo por brindarle el marco institucional a este proyecto. Muchas gracias al Grupo de Ingeniería Biológica del Centro Universitario Regional Litoral Norte y a la Unidad de Bioinformática del Institut Pasteur por acercarme los insumos, infraestructura y herramientas para llevar a cabo este proyecto.

Muchas gracias la Comisión Académica de Posgrado, por los fondos dedicados a mi formación y este proyecto.

A mis tutores Hugo, Juan y Víctor por la guía, paciencia, apoyo, confianza y oportunidades.

A Martín por toda la ayuda brindada en este proyecto y hacer que todo funcione.

A Lucía por las ideas, el apoyo y por abrir caminos.

Al grupo de Ingeniería Biológica, por ser hogar, familia y aprendizaje constante.

A la UBi, por abrirme sus puertas, adoptarme y ser el sitio cálido al que siempre queremos volver.

A mi familia y amigos de siempre. A mamá, papá, Pauli y Germán.

Índice general

Agradecimientos iii

1. Introducción 1
1.1. Motivación . . . 1
1.2. Fundamentos . . . 2
1.2.1. Enfermedades raras . . . 2
1.2.2. Patrones de herencia . . . 3
1.2.3. Variantes en el genoma humano . . . 6
1.2.3.1. Tipos de variantes . . . 6
1.2.4. Impacto de las variantes . . . 8
1.2.5. Identificación e interpretación de variantes . . . 11
1.2.5.1. Interpretación y clasificación de variantes . . . 13
1.3. Objetivos . . . 16
1.3.1. Objetivo general . . . 16
1.3.2. Objetivos específicos . . . 16
1.4. Esquema de trabajo general . . . 17
1.5. Estructura de la documentación . . . 18

2. Antecedentes 21
2.1. Herramientas de clasificación usando *Machine Learning* . . . 23

3. Materiales y métodos comunes 29
3.1. ClinVar . . . 29
3.2. Anotación de características biológica . . . 33
3.2.1. Bases de Datos interrogadas . . . 34

4. Plataforma de aprendizaje 37
4.1. Especificaciones del Sistema Web . . . 37
4.1.1. Alcance . . . 37
4.1.2. Requerimientos funcionales . . . 38
4.1.3. Requerimientos no funcionales . . . 38
4.1.4. Actores . . . 39
4.1.5. Casos de uso . . . 39
4.1.6. Modelado de dominio . . . 41
4.2. Diseño del Sistema Web . . . 42
4.2.1. Arquitectura . . . 42
4.2.2. Elección de tecnologías . . . 43
4.2.2.1. Lenguaje de programación . . . 43
4.2.2.2. Front-end . . . 43
4.2.2.3. Backend . . . 49
4.3. Implementación . . . 53
4.3.1. Flujo de usuario . . . 53
4.3.1.1. Proceso común de clasificación de variantes . . . 57

4.3.1.2. Datos . . . 58
4.3.1.3. Tipos de usuarios . . . 59
4.3.2. Interfaz de usuario y navegación . . . 60
4.3.2.1. Selección de variante . . . 62
4.3.2.2. Interpretación de la variante y evidencia . . . 63
4.3.2.3. Desarrollo de la interfaz en Dash . . . 72
4.3.3. Backend . . . 73
4.3.3.1. Base de Datos . . . 73
4.3.3.2. API . . . 77
4.3.3.3. Lógica de transformación de datos . . . 77
4.4. Resultados preliminares y discusiones . . . 82
4.4.1. Disponibilidad de la aplicación . . . 82
4.4.2. Estado actual . . . 83
4.4.3. Evaluación general de la funcionalidad . . . 84
4.4.4. Discusión general . . . 90
4.5. Conclusiones y trabajo futuro . . . 92

5. Clasificación de variantes **95**
5.1. Introducción . . . 95
5.1.1. Objetivos . . . 95
5.1.1.1. Objetivo general . . . 95
5.1.1.2. Objetivos específicos . . . 95
5.2. Metodología . . . 96
5.2.1. Pre-procesamiento de datos . . . 96
5.2.1.1. Descripción y preparación de datos . . . 96
5.2.2. Manipulación de valores faltantes . . . 100
5.2.3. Análisis de características . . . 102
5.2.4. Conjuntos de entrenamiento, validación y *test* . . . 102
5.2.5. Entrenamiento y selección de modelos . . . 103
5.2.6. Implementación . . . 105
5.2.7. Integración . . . 105
5.3. Resultados y discusión . . . 105
5.3.1. Imputación de valores faltantes . . . 105
5.3.2. Análisis de características . . . 108
5.3.3. Entrenamiento . . . 109
5.4. Conclusiones y trabajo futuro . . . 113

6. Conclusiones finales **115**

A. Reglas ACMG **117**
A.1. Criterios de clasificación de las variantes patogénicas . . . 118
A.2. Criterios de clasificación de las variantes benignas . . . 119
A.3. Reglas de combinación de criterios . . . 120

B. Datos **121**
B.1. Formato VCF . . . 121

C. Plataforma web **123**
C.1. Base de Datos . . . 124
C.2. Evaluación . . . 125
C.2.1. Metodología de evaluación . . . 125
C.2.1.1. Experimento 1: Definición de información a mostrar . . . 125

C.2.1.2. Experimento 2: Organización de contenido y presentación de la variante 125
C.2.1.3. Experimento 3: Ingreso de nuevas variantes a la plataforma 125
C.2.1.4. Experimento 4: Evaluación de paciente para comparar con planilla . . 126
C.2.1.5. Experimento 5: Evaluación de plataforma de etiquetado 126
C.2.1.6. Experimento 6: Evaluación de plataforma inicial de aprendizaje . . . 126
C.3. Evaluación de experiencia de usuarios preliminar 127

Bibliografía **131**

Índice de figuras

1.1. Ejemplos de modos de herencia representados través de diagramas de *pedigree*. 5
1.2. Esquema en representación de algunos tipos de variantes identificadas en el genoma humano. 7
1.3. Ejemplo de efecto de SVN sobre un codón, clasificada por su impacto en la secuencia proteica. 9
1.4. Ejemplo de una *frameshift insertion* representada de forma gráfica. 10
1.5. *Pipeline* de procesamiento de datos de secuenciación (WGS o WES) para la detección de variantes. 12
1.6. Organización de los criterios ACMG/AMP por tipo de evidencia y peso. 16
1.7. Esquema general del trabajo realizado. 18

3.1. Flujo de funcionamiento de ANNOVAR. 35

4.1. Diagrama UML de casos de uso de la plataforma. 39
4.2. Modelo de dominio de la plataforma desde el punto de vista de los datos. 41
4.3. Arquitectura de la aplicación web. 42
4.4. Comparación de popularidad de herramientas de *dashboarding* 45
4.5. Arquitectura general de una aplicación Dash. 48
4.6. Flujo general de navegación de los usuarios en la plataforma web. 54
4.7. Flujo detallado de la navegación de los usuarios en la plataforma web. 55
4.8. Esquema general del proceso de clasificación de variantes. 57
4.9. Esquema de interacción entre los tipos de variantes y usuarios según su funcionalidad en la plataforma. 59
4.10. Mapa de navegación de los usuarios en la interfaz web. 61
4.11. Selección inicial del tipo de variante a trabajar en el proceso de etiquetado. 62
4.12. Ejemplo de selección del método de obtención de una variante para su clasificación. . . 63
4.13. Vista de la información general en la interfaz. 65
4.14. Vista de la información de OMIM para una variante. 66
4.15. Ejemplo de vista de frecuencias alélicas en la plataforma. 68
4.16. Ejemplo de vista de predictores *in silico* en la plataforma. 69
4.17. Ejemplo de vista de reglas ACMG en la plataforma. 71
4.18. Diagrama de la base de datos 75

5.1. Diagrama del proceso de *k-fold crossvalidation*. 104
5.2. Diagrama de barras representando la completitud del conjunto de datos. 107
5.3. Dendrograma que representa la correlación entre las variables dados los patrones de valores faltantes. 107
5.4. Correlación de variables finales. 109
5.5. Matrices de confusión de los modelos finales obtenidos para la clasificación de 5 clases: B(1), LB(2), VUS(3), LP(4), P(5) 112

B.1. Estructura de un archivo VCF. 121

C.1. Diagrama EER de la base de datos. 124

Índice de cuadros

1.1. Resumen de los tipos más grandes de variación en el genoma humano, con su tamaño y número por genoma . . . 7

3.1. Nomenclatura para el significado clínico y su correspondiente definición. . . . 31
3.2. Tabla de combinación de valores de significado clínico y su reporte correspondiente . . 32
3.3. Estado de revisión aportado por ClinVar con su correspondencia en número de estrellas en la página web [149] [150]. . . . 33

4.1. Tabla comparativa entre *frameworks* de *dashboarding* para la generación de interfaces web. . . . 47
4.2. Tabla comparativa entre DBMS relacional y no relacional de acuerdo a los objetivos del proyecto. . . . 52
4.3. Información presentada en la interfaz tomada de fuentes externas. . . . 64
4.4. Código de colores para rangos de frecuencias alélicas. . . . 68
4.5. Resumen de los tipos de variantes asignados a cada nivel de usuario en la instancia de aprendizaje. . . . 79
4.6. Correspondencia entre nivel de usuario y rango de XP. . . . 80
4.7. Distribución de los usuarios autenticados en la plataforma de acuerdo a la institución participante. . . . 83
4.8. Distribución de variantes adquiridas de ClinVar en la base de datos según su tipo y la funcionalidad a la que aportan. . . . 84
4.9. Tiempos de respuesta de la página ante determinadas tareas. . . . 85

5.1. Distribución inicial de tipos de variantes según su significado clínico o etiqueta de patogenicidad para clasificación. . . . 96
5.2. Características consideradas inicialmente para el entrenamiento de modelos. . . . 100
5.3. Codificación de valores categóricos a numéricos. . . . 100
5.4. Distribución de tipos de variantes usadas en el entrenamiento, posterior a aplicación de filtros por valores faltantes. . . . 108
5.5. Cantidad de variantes seleccionadas para los conjuntos de entrenamiento y prueba. . . . 109
5.6. Desempeño de los clasificadores evaluados inicialmente. . . . 110

A.1. Lista de criterios para la clasificación de variantes patogénicas. . . . 118
A.2. Lista de criterios para la clasificación de variantes benignas. . . . 119
A.3. Reglas de combinación de criterios para clasificar variantes. . . . 120

C.1. Resumen de las respuestas adquiridas en los cuestionarios realizados a los usuarios. . . . 130

Lista de abreviaciones

ACMG	**A**merican **C**ollege of **M**edical
ADN	**Á**cido **D**esoxirribo**N**ucleico
API	**A**pplication **P**rogramming **I**nterface
ARN	**Á**cido **R**ibo**N**ucleico
ARNm	**Á**cido **R**ibo**N**ucleico **m**ensajero
CENUR LN	**Cen**tro **U**niversitario **R**egional **L**itoral **N**orte
CNN	**C**onvolutional **N**eural **N**etwork
CNV	**C**opy **N**umber **V**ariant
CROWDLAB	**C**lassifier **R**efinement **O**f cro**WD**sourced **LAB**els
CRUD	**C**reate**R**ead **U**pdate **D**elete
CSS	**C**ascading**S**tyle **S**heets
DBC	**D**ash**B**ootstrap **C**omponents
DBMS	**D**ata**B**ase **M**anagement **S**ystem
DCC	**D**ash**C**ore **C**omponents
DL	**D**eep **L**earning
DNN	**D**eep **N**eural **N**etwork
EER	**E**nhaced **E**ntity **R**elationship
ER	**E**nfermedad **R**ara
ESP	**E**xome **S**equencing **P**roject
ExAC	**Ex**ome **A**ggregation **C**onsortium
gnomAD	**genome A**ggregation **D**atabase
GBDT	**G**radient **B**oosting **D**ecision **T**ree
HGMD	**H**uman **G**enome **M**utation **D**atabase
HGNC	**HUGO G**ene **N**omenclature **C**ommittee
HMM	**H**idden **M**arkov **M**odel
HTML	**H**yper**T**ext **M**arkup **L**anguage
HTTP	**H**ypertext **T**ransfer **P**rotocol
HUGO	**H**uman **G**enome **O**rganization
INDEL	**I**nsertion or **D**eletion
KNN	**K**-**N**earest **N**eighbor
LFS	**L**arge **F**ile **S**torage
MAF	**M**inor **A**llele **F**requency
MICE	**M**ultivariate **I**mputation by **C**hained **E**quations
MVP	**M**inimum **V**iable **P**roduct
NCBI	**N**ational **C**enter for **B**iotechnology **I**nformation
NGS	**N**ext **G**eneration **S**equencing
NLP	**N**atural **L**anguage **P**processing
OMIM	**O**nline **M**endelian **I**nheritance in **M**an
PoC	**P**roof **O**f **C**oncept
REST	**RE**presentational **S**tate **T**ransfer
SNP	**S**ingle **N**ucleotide **P**olimorphysm
SNV	**S**ingle **N**ucleotide **V**ariant

SV	**S**tructural **V**ariants
SVM	**S**upport **V**ector **M**achine
UML	**U**nified **M**odeling **L**anguage
VCF	**V**ariant **C**all **F**ormat
VUS	**V**ariant of **U**ncertain **S**ignificance
WES	**W**hole **E**xome **S**sequencing
WGS	**W**hole **G**enome **S**sequencing
XP	E**X**perience **P**oints

Capítulo 1

Introducción

1.1. Motivación

Las variantes genómicas, que hacen referencia a variaciones en la secuencia de ADN de un organismo, resultan fundamentales para comprender, en parte, la base las enfermedades genéticas. Los avances en las tecnologías genómicas moleculares, como las tecnologías de secuenciación de nueva generación (NGS), han ayudado al estudio del genoma humano para el diagnóstico y la evaluación del riesgo de un amplio conjunto de enfermedades [1] [2]. Un campo en el que la aplicación de estas tecnologías han tenido un gran impacto es en el del diagnóstico de enfermedades raras: patologías poco frecuentes, que afectan a menos de 1 cada 2000 personas [3]. Dada su baja frecuencia y la heterogeneidad de sus síntomas, la obtención de un diagnóstico para estas enfermedades se transforma en una verdadera "odisea", en la que el paciente espera en promedio 5 años hasta ser diagnosticado, lo que representa el primer paso para recibir una atención médica adecuada. El hecho de que aproximadamente el 80 % de estas enfermedades tienen un origen genético, ha permitido que las herramientas de NGS aporten en la mejora de la tasa diagnóstica de estas enfermedades, reduciendo el tiempo de espera por un diagnóstico. Es por ello que secuenciación del genoma y/o exoma humano se ha convertido en una herramienta frecuente en la práctica clínica en un amplio rango de enfermedades de base genética. Sin embargo, la tasa de diagnóstico actual, que oscila entre el 28 % en la secuenciación de exoma completo (WES[1]) y el 57 % en los estudios más exhaustivos de WGS (WGS[2]), aún dista de ser satisfactoria, considerando que aproximadamente para un 50 % de enfermedades la causa no puede ser determinada con certeza [4].

Los estudios de WES y WGS, y las mejoras en las tecnologías de los mismos han permitido facilitar la determinación de un gran número de variantes genómicas, y con ello evidencian una de las tareas más complejas y desafiantes en el campo de la genómica médica: la evaluación de la patogenicidad de las variantes genéticas identificadas. En el contexto de los trastornos hereditarios, sólo unas pocas de las variantes identificadas podrían ser patogénicas o causales de la enfermedad de interés, y en un escenario en el que cada vez se genera una mayor cantidad de datos, el análisis y la interpretación de las variantes genómicas se han vuelto cada vez más complejo y desafiante. Para lograr obtener un diagnostico, las variantes identificadas tienen que ser evaluadas en función de metadatos obtenidos a partir de procesos de anotación (determinación del efecto funcional de las variantes), que forma parte del flujo de trabajo bioinformático. Para ello se utiliza información de varias bases de datos asociadas al genoma humano y mutaciones en el mismo. Con esta información obtenida de procesos de anotación, las variantes se pueden clasificar según su efecto en variantes patogénicas, benignas, probablemente patogénicas o benignas, variantes de significado incierto (VUS) y hallazgos incidentales. Para lograr esta clasificación, se utilizan lineamientos estandarizados basados en una combinación de criterios de expertos, datos empíricos y los datos adquiridos de distintas fuentes. Estos lineamientos buscan resolver las discrepancias entre laboratorios con diferentes protocolos y la clasificación de variantes con evidencia contradictoria. Si bien estas reglas y guías aportaron mayor consistencia

[1] *Whole Exome Sequencing*, en inglés
[2] *Whole Genome Sequencing*, en inglés

en la clasificación, aún hay varios aspectos con poca especificidad, y mientras algunos factores pueden ser cuantificados, y por ende, se puede automatizar la clasificación, otros factores dependen del caso, y requieren evaluación humana [5]. Es decir, hay un alto grado de juicio de expertos, y una curva de aprendizaje en la aplicación precisa de estas reglas, generando un proceso de clasificación con nuevas discrepancias [6]. El proceso de caracterizar la relevancia clínica o interpretación de una variante particular en las categorías más determinantes como patogénica (causa de enfermedad) o benigna (no causante de enfermedad), plantea un desafío debido a cuestiones como: diferencias en la información presente en los flujos de trabajo seguidos por cada laboratorio o grupo de trabajo, disponibilidad limitada de recursos computacionales, o falta de de profesionales capacitados. Todo lo anterior ocurre aún a pesar de que se han desarrollado varios algoritmos computacionales orientados a la predicción del impacto clínico, basados en características como la homología, la conservación evolutiva, la función de las proteínas codificadas por el gen en cuestión, entre muchas otras [7]. A pesar de los grandes esfuerzos realizados en este sentido en los últimos años, asociados a la creciente capacidad de cómputo, la interpretación de estas variantes sigue siendo un gran desafío. Aunque ha aumentado el desarrollo de reglas y herramientas para la interpretación de variantes, muchas variantes permanecen sin clasificar o con una interpretación contradictoria de su patogenicidad [8]. Varios estudios han identificado inconsistencias en la clasificación de variantes entre laboratorios, bases de datos de referencia e incluso dentro de bases de datos de genes específicos [6, 9, 10, 11].
Para hacer frente a estos retos, se han desarrollado distintos tipos de herramientas bioinformáticas que ayudan a clasificar las variantes genómicas. La clasificación de variantes puede ser modelada como un problema de aprendizaje automático, y en este aspecto se basan dichas herramientas, las cuales utilizan diversos algoritmos y técnicas de aprendizaje automático para predecir el impacto funcional de las variantes genómicas en la expresión génica, la función proteica y la enfermedad. Sin embargo, la precisión y sensibilidad de estas herramientas se ven limitadas por la falta de bases de datos exhaustivas y curadas de variantes genómicas, así como por la falta de estandarización en la clasificación de variantes [12].
Por lo expuesto anteriormente, se puede decir que el proceso de diagnóstico requiere no solo de recursos humanos formados en los estándares requeridos para la clasificación, sino también de la potencia de herramientas computacionales sólidas que permitan asistirlo. Teniendo en cuenta las limitaciones provenientes de ambos aspectos, la motivación de este trabajo se basa en atacarlos, a través del desarrollo de herramientas que asistan a profesionales en la clasificación de variantes genómicas, no solamente desde la clasificación automática de las mismas, sino también desde la guía del usuario en el proceso de evaluación. El presente proyecto de tesis, está orientado al procesamiento de variantes genómicas en general, pero está motivado por el interés de seguir aportando en distintas estrategias para mejorar las posibilidades de diagnostico de enfermedades raras. ¿El objetivo final? Ir más allá del 50 % actual de la tasa diagnóstica. Para lograrlo, se empieza por pasos pequeños, y uno de ellos, implica la realización de evaluaciones e implementaciones preliminares de posibles enfoques. Este trabajo a se enmarca en el contexto de un estudio profundo y prolongado en el tiempo que representa una de las líneas de trabajo de la Unidad de Bioinformática (UBi) del Instituto Pasteur de Montevideo (IPMon) en genómica médica, en donde se realizó este trabajo. Desde el año 2015, parte de la UBi en conjunto con otros grupos de trabajo, se encuentra desarrollando el proyecto URUGENOMES [13], cuya tercera fase se encuentra aplicada al estudio del genoma de 30 pacientes con enfermedades raras o de difícil diagnóstico, de relevancia en nuestro país [14, 15, 16].

1.2. Fundamentos

1.2.1. Enfermedades raras

Las “enfermedades raras” (ER) también conocidas como “enfermedades huérfanas”, son un grupo de trastornos que afectan a un pequeño porcentaje de la población. El término “raro” puede atribuirse a distintos contextos de prevalencia, y las definiciones son adoptadas por distintos formadores

de políticas nacionales, también dependiendo del tamaño de la población. Entre las definiciones más aceptadas se encuentra la de Estados Unidos, en donde una enfermedad se considera rara si afecta a menos de 200.000 individuos, mientras que en Europa se considera rara si afecta a menos de 1 de cada 2.000 personas [17, 18]. A los efectos de este trabajo, se utilizará la definición establecida por *Orphanet*, en la cual se define a una ER como aquella que afecta a menos de 1 en 2000 personas [17]. Mientras cada ER puede afectar sólo a un pequeño número de individuos, colectivamente hay más de 6.000 ER que afectan a millones de personas en todo el mundo. Un trabajo referencia en números y frecuencias en ER publicado en 2019 estimó que las ER al momento se encontraban afectando al 3.5 % - 5.9 % de la población mundial, equivaliendo en aquel momento a una estimación de 263-446 millones de personas en todo el mundo [19].

Las ER son condiciones clínicas heterogéneas, progresivas, discapacitantes, crónicamente debilitantes y/o potencialmente mortales, y en la mayoría de los casos afectan a niños (un 70 % de las ER comienzan en la infancia) [19, 20]. A pesar de su diversidad, tienen en común una serie de dificultades asociadas características:

- La "odisea diagnóstica": la mayoría de las ER tienen síntomas inespecíficos que pueden confundirse con afecciones más comunes, lo que provoca retrasos en el diagnóstico y el tratamiento. En promedio se tarda alrededor de 5 años en brindar un diagnóstico acertado, tiempo en el que el paciente, la familia y el personal de la salud debieron enfrentarse a una verdadera odisea (de allí el nombre), caracterizada por un gran desgaste físico, emocional y económico [21, 22].
- Falta de conocimientos: Las ER suelen ser poco conocidas por los profesionales sanitarios debido a su rareza, lo que puede llevar a un diagnóstico erróneo o a un tratamiento inadecuado [21].
- Opciones de tratamiento limitadas: Debido a la reducida población de pacientes, suele haber una falta de investigación en el desarrollo de tratamientos para las ER. Se estima que más del 90 % de las enfermedades raras no tienen actualmente un tratamiento eficaz. Incluso cuando existen tratamientos, pueden ser prohibitivamente caros o de difícil acceso [21].
- Aislamiento del paciente: Las personas con ER a menudo se sienten aisladas debido a la falta de concienciación y apoyo a su enfermedad. Esto puede tener un impacto significativo en su salud mental y calidad de vida en general [23].
- Carga económica: El costo del tratamiento de una ER puede ser extremadamente alto, tanto para el paciente como para la sociedad en su conjunto. Esto puede acarrear dificultades económicas para las personas y sus familias, así como tensiones en los sistemas sanitarios [24, 25].
- Acceso a la asistencia: Debido a la rareza de su enfermedad, las personas con ER pueden tener un acceso limitado a la atención especializada y a los servicios de apoyo, lo que puede agravar aún más los retos a los que se enfrentan [26].

Si bien la causa de las enfermedades raras en general es desconocida, se estima que aproximadamente un 80 % tienen origen genético y la mayoría de ellos presentan una distribución familiar compatible con un origen monogénico o mendeliano [19, 27]. De allí el interés por aplicar las técnicas NGS en el diagnóstico de este tipo de enfermedades. Muchas enfermedades no diagnosticadas se han identificado mediante la secuenciación del exoma, que examina las regiones codificadoras de proteínas, las cuales constituyen menos del 2 % del genoma[28].

1.2.2. Patrones de herencia

Muchas afecciones y enfermedades dependen del genotipo de un único locus (o gen) y su herencia sigue las leyes de Mendel de segregación, selección independiente y dominancia. Hasta la fecha, se han

identificado más de 8000 fenotipos cuya base molecular se conoce los cuales se encuentran reportados en la base de datos OMIM junto a los genes asociados [29, 30].
La base para entender la segregación genética es que un organismo diploide contiene dos copias de cada gen (excepto aquellos que se encuentran en los cromosomas sexuales). Una copia de cada gen es heredada de cada progenitor. Los rasgos pueden transmitirse como autosómicos o ligados al sexo, y como dominantes o recesivos. Con frecuencia, estos alelos no son idénticos. Una persona portadora de dos copias idénticas del mismo alelo en ambos autosomas es homocigota para este alelo, mientras que una persona portadora de dos alelos diferentes es heterocigota para el locus. Un alelo dominante es aquel que da lugar a un fenotipo concreto independientemente de si el segundo alelo es "normal" o no; en este caso el segundo alelo no puede compensar el efecto del alelo dominante. Por otro lado, un alelo recesivo, en cambio, no provoca un fenotipo por sí solo; en este caso, el alelo "normal" es suficiente o puede compensarlo. En este caso, los rasgos que se expresan mediante herencia mendeliana recesiva se expresan solo en individuos que poseen dos copias de alelos mutantes, heredados de de ambos progenitores, y que por lo tanto son homocigotas para el alelo recesivo [30, 31]. Teniendo en cuenta los conceptos mencionados anteriormente, pueden distinguirse cinco tipos distintos de patrones de herencia mendeliana:

- Autosómico dominante: en este patrón de herencia una sola copia de un gen anormal en uno de los autosomas (cromosomas no sexuales) es suficiente para causar un trastorno o rasgo genético concreto. En este caso, si uno de los progenitores es portador del alelo afectado, existe un 50 % de probabilidad de que cada uno de sus descendientes herede el gen y, por tanto, desarrolle el trastorno o rasgo genético. El trastorno o rasgo suele aparecer en todas las generaciones de una familia afectada, siempre que el gen anormal esté presente, afectando por igual a ambos sexos [30]. Algunos ejemplos de trastornos causados por herencia autosómica dominante son la enfermedad de Huntington [32] (Figura 1.1) o Neurofibromatosis tipo 1 [33], entre otros.

- Autosómico recesivo: en este patrón de herencia se requieren dos copias de un gen determinado, una de cada progenitor, para expresar un rasgo o trastorno concreto. Esto significa que significa que un individuo debe heredar una copia mutada del gen de cada progenitor para presentar el rasgo o trastorno. Normalmente,los individuos afectados en este tipod e herencia tienen dos progenitores no afectados, portadores de un solo alelo patogénico. En la herencia autosómica recesiva, hay un 25 % de probabilidad de que la descendencia herede dos copias del gen mutado, un 50 % de probabilidad de heredar una copia del gen mutado y ser portador, y un 25 % de probabilidades de no heredar ninguna copia del gen mutado. Ambos sexos tienen las mismas probabilidades de heredar el gen mutado, no obstante la incidencia se ve frecuentemente incrementada en familias donde los padres son consanguíneos. Los trastornos autosómicos recesivos están causados por mutaciones en genes localizados en cromosomas autosómicos [30]. Algunos ejemplos de trastornos autosómicos recesivos son la Fibrosis Quística [34] y Anemia falciforme [35] (Figura 1.1), entre otros.

- Recesivo ligado al comosoma X: en este modo de herencia el gen responsable de un rasgo o trastorno concreto se localiza en el cromosoma X, uno de los dos cromosomas sexuales. Por lo tanto las enfermedades causadas por variantes recesivas en loci situados en el cromosoma X afectan de forma diferente a mujeres y hombres. Dado que las mujeres tienen dos cromosomas X, pueden ser portadoras de dos copias del gen, mientras que los hombres sólo tienen un cromosoma X y, por tanto, sólo una copia del gen. En este modo un varón que hereda una copia mutada del gen en su cromosoma X expresará el rasgo o trastorno, ya que no tiene otra copia del gen para enmascarar sus efectos. Las mujeres que heredan una copia mutada del gen se denominan portadoras (tienen una copia funcional del gen y una copia mutada), pero no expresan el rasgo o trastorno. Las mujeres portadoras pueden transmitir el gen mutado a sus hijos, incluidos los varones, quienes tienen un 50 % de probabilidad de heredar el gen mutado y expresar el rasgo o trastorno. Algunos ejemplos de trastornos recesivos ligados al cromosoma X incluyen

la hemofilia A y B [36] (Figura 1.1), el daltonismo rojo-verde [37] y la distrofia muscular de Duchenne [38], entre otros. Estos trastornos afectan predominantemente a los varones.

- Dominante ligado al comosoma X: en este patrón de herencia, si una persona hereda el cromosoma X mutado de cualquiera de sus progenitores, desarrollará el trastorno o la enfermedad asociada. En este caso, todas las hijas de un varón afectado heredarán la enfermedad, mientras que todos sus hijos varones no estarán afectados. En el caso de una mujer afectada, si tiene una única variante patogénica, sus hijos tendrán un 50 % de probabilidad de estar afectados. Sin embargo, si ha heredado una variante patógena de cada uno de sus padres, lo normal es que ambos progenitores estén afectados y que todos sus hijos también lo estén. Si bien los trastornos dominantes ligados al cromosoma X son raros, algunos ejemplos son el síndrome de Rett [39] y el síndrome del cromosoma X frágil [40], entre otros.

- Ligado al comosoma Y: este modo implica genes localizados en el cromosoma Y, otro de los dos cromosomas sexuales. Los rasgos ligados al cromosoma Y se heredan exclusivamente de padres a hijos, ya que las hijas no heredan un cromosoma Y. Dado que gran parte del cromosoma Y existe en estado hemicigótico, no se aplican las definiciones recesivo y dominante; como tal, el fenotipo de las variantes del cromosoma Y será manifiesto. Los rasgos ligados al cromosoma Y se heredan de forma estrictamente vertical, lo que significa que se transmiten sin cambios de una generación a la siguiente, excepto en el caso poco frecuente de *de novo*. Los rasgos ligados al Y son relativamente raros, ya que el cromosoma Y sólo contiene un pequeño número de genes, la mayoría de los cuales están implicados en la determinación del sexo masculino y la fertilidad [30].

En la Figura 1.1 se presentan algunos ejemplos de modos de herencia representados de forma gráfica a través de diagramas de *pedigree* [41].

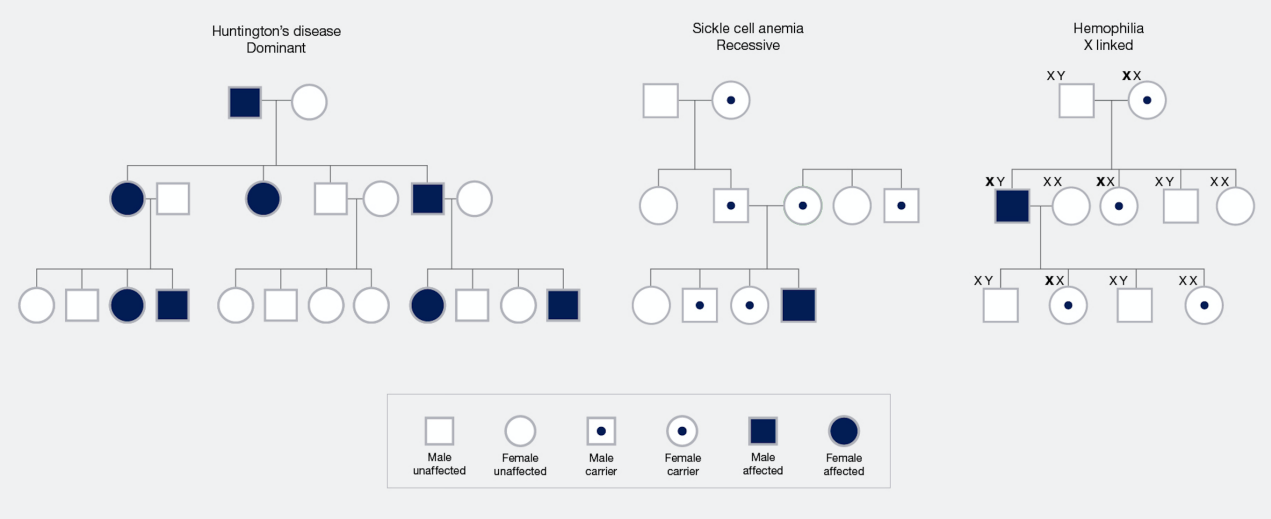


FIGURA 1.1: Ejemplos de modos de herencia representados través de diagramas de *pedigree*. Para las representaciones se usan tres ejemplos de enfermedades que cumplen distintos patrones de herencia: enfermedad de Huntington (autosómica dominante), anemia falciforme (autosómica recesiva) y hemofilia (recesiva ligada al cromosoma X). Extraído de [41].

Es importante tener en cuenta que las herramientas desarrolladas en el presente trabajo implican el análisis de variantes que responden a los modos de herencia presentados anteriormente. Esto se fundamenta en que la cigosidad es un elemento central en la interpretación de una variante en función de su modo de herencia.

1.2.3. Variantes en el genoma humano

El genoma humano es la secuencia de ADN completa de ser humano que comprende aproximadamente 3×10^9 de pares de bases. La mayor parte de dicha secuencia de ADN se encuentra concentrada en el núcleo de cada célula en forma de 22 pares de cromosomas autosómicos y dos cromosomas sexuales X e Y, representando lo que se conoce como genoma nuclear. Además hay una pequeña cantidad de ADN en las mitocondrias, consistiendo en el genoma mitocondrial. Está organizado en diferentes regiones, y su contenido se puede dividir en dos grandes partes: secuencia de ADN codificante y no codificante. La porción codificante conocida como exoma comprende alrededor del 1-2 % del total del genoma, y contiene las secuencias de ADN que codifican para moléculas de ARN mensajero, luego traducidas a proteínas: elementos esenciales para la estructura, función y regulación de los tejidos y órganos del cuerpo [30]. El genoma humano contiene aproximadamente 20000 genes codificantes de proteínas. Por otro lado, aproximadamente un 98 % del genoma se encuentra compuesta por secuencias no codificantes de diferentes tipos. Entre esas secuencias se encuentran unos 22.000 genes que codifican los distintos tipos de ARN funcional, intrones (aproximadamente un 25 % del genoma), pseudogenes, secuencias codificantes para regiones no traducidas del ARN mensajero, secuencias reguladoras, secuencias de ADN repetitivas (aproximadamente un 50 % del genoma) y secuencias relacionadas con elementos móviles o transponibles [30, 42, 43]. El genoma humano representa el conjunto completo de instrucciones necesarias para construir y mantener un organismo, incluidas las instrucciones para fabricar proteínas y regular su actividad. No existe una "secuencia de ADN humano" canónica [31]. Se estima que un 99.9 % de la secuencia del genoma humano es idéntica entre los individuos, mientras que el resto de la secuencia consiste variaciones que se generan constantemente, tanto a nivel somático (en la proliferación de células en los tejidos) como germinal (heredados de padres a hijos). Las variaciones en la secuencia de bases se dan de un individuo a otro, en promedio cada pocos cientos de bases. Estas variaciones en la secuencia pueden tener distintos efectos: pueden ser responsables de explicar las diferencias físicas entre los individuos, pueden tener un efecto neutro o resultar en una ventaja selectiva, o pueden ser causantes de trastornos médicos.

El término "variante" se refiere a cualquier alteración heredable de la secuencia o la estructura del ADN entre individuos o poblaciones. Las variantes de la secuencia de ADN pueden producirse a muchos niveles, desde un solo nucleótido hasta un cromosoma entero. Estas variaciones pueden recibir distintos nombres, entre los que se encuentran los términos mutación o polimorfismo. El termino mutación se refiere a un cambio permanente en la secuencia de nucleótidos. Por otro lado, el término polimorfismo se refiere a aquellos cambios en la secuencia que son frecuentes en la población, y tienen una frecuencia en la misma que supera el 1 %. En general, se tiende a utilizar los términos "mutación" y "polimorfismo" en asociación a efectos deletéreos o no deletéreos, respectivamente, sobre la secuencia. Para uniformizar criterios, es que se ha recomendado fuertemente la sustitución de ambos términos por el de "variante" [44].

1.2.3.1. Tipos de variantes

El espectro de la variación genética humana es amplio y abarca desde mutaciones puntuales hasta grandes aneuploidías cromosómicas que afectan a cromosomas enteros. Los tipos de variantes se pueden dar a distintas frecuencias dentro del genoma, y pueden contribuir de distintas formas, a por ejemplo, el desarrollo de una enfermedad. A continuación serán descritos los tipos de variantes más representativos.

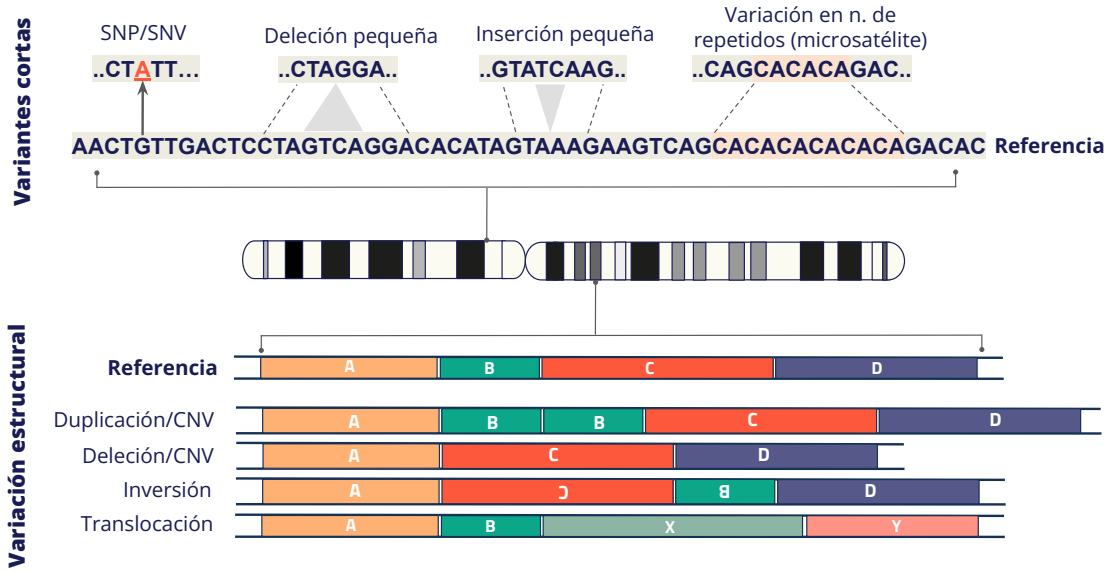


FIGURA 1.2: Esquema en representación de algunos tipos de variantes identificadas en el genoma humano. Modificado de [30].

Tipo	Tamaño (pb)	Número por genoma
Variantes de un solo nucleótido (SNV)	1	$\approx 3{,}5 \times 10^6$
Inserciones/deleciones (INDEL) pequeñas	1–49	$\approx 4{,}5 \times 10^5$
Repeticiones en tandem (STR) cortas	1–6	$\approx 1 \times 10^5$
Número variable de repeticiones en tándem (VNTR)	7-49	-
Variantes estructurales (SV)	≥ 50	$\approx 1 \times 10^5$

CUADRO 1.1: Resumen de los tipos más grandes de variación en el genoma humano, con su tamaño y número por genoma

Variantes de un solo nucleótido y pequeñas variantes de inserción o deleción

Las variantes de un solo nucleótido (SNV) y las pequeñas variantes de inserción/deleción (INDELs) (menores a 50 pb), o variantes cortas[3], constituyen la gran mayoría de las variantes en la población humana. Los SNV (que incluye a polimorfismos de nucleótido único (SNP)) son los cambios que afectan a un solo par de bases. Consisten en la sustitución de un nucleótido, ya sea una transición, en la que se intercambian nucleótidos con un base purina (A/G) o con base pirimidina (C/T), o una transversión en la que se intercambia un nucleótido con base purina por uno con base pirimidina. Una comparación con un genoma referencia arrojaría aproximadamente ≈3-4 millones de SNVs en un humano [45, 46].

[3]El término variantes cortas será usado a lo largo del trabajo refiriéndose a SNVs o INDELs con un tamaño inferior a 50 pb

Por otro lado, las inserciones consisten en la pérdida de un nucleótido puntual o de un fragmento de ADN, y a las inserciones la inserción de un nucleótido puntual o de un fragmento de ADN. En ambos casos, la inserción o deleción de un fragmento, se considera una variante corta cuando la misma consiste en menos de 50 pb. Se estima que hay aproximadamente ≈0.4-0.5 millones de INDELS en una comparación típica de un genoma humano frente a la referencia.
Muchos loci presentan una frecuencia alta en la variación individuo a individuo. Este tipo de variantes se conocen como polimorfismos, los que se definen formalmente como un locus con dos o más alelos en el cual al menos uno de los alelos menos comunes tiene una frecuencia igual o mayor al 1 %. Mientras que la gran mayoría estas variantes no tienen impacto funcional a nivel molecular o fenotípico, cada genoma tiene más de 100 variantes de truncamiento de proteínas que introducen un codón de parada prematuro, y más de 20 de ellas son raras en la población humana y potencialmente deletéreas [46].

Variantes estructurales

La variación estructural es una categoría que se ha utilizado para referirse colectivamente a las diferencias de al menos 50 pb de longitud entre dos genomas individuales. Las variantes estructurales (SV[4]) incluyen translocaciones, inversiones, grandes INDELS, variantes en el número de copias (CNV[5]) e inserción de elementos móviles [30, 46]. Un genoma humano típico tiene ≈ 10,000 SV. Este tipo de variantes son poco numerosas en comparación con las variantes cortas, pero suelen tener consecuencias más graves en promedio, debido a su tamaño e impacto. Éstas pueden actuar ejerciendo efectos funcionales cambiando la dosis génica, alterando la función génica o reorganizando elementos reguladores y/o genes para alterar el contexto genómico [46]. Es de esperar que estas variantes grandes que eliminan o duplican varios genes o incluso cromosomas enteros suelen tener efectos fenotípicos drásticos y no se observan en la mayoría de los individuos. En general las formas más pequeñas y prevalentes de variantes estructurales suelen afectar sólo a uno o unos pocos genes o se encuentran en regiones no codificantes. Puntualmente, las SV no serán atacadas en este trabajo, por lo que no se profundizará más adelante sobre las mismas.

1.2.4. Impacto de las variantes

El efecto de los cambios a nivel de la secuencia depende de dónde la variante se encuentre localizada: en un exón, una región de control, un intrón, y en el caso de variantes más grandes, si abarcan genes completos o grupos de genes. Además de depender de su localización, el efecto/impacto va a depender del tipo de variante [31].
Si se produce un SNV en la región codificante de un gen, pueden darse varios resultados. Por un lado puede provocar cambios silenciosos o sinónimos que provocan el cambio de un codón por otro diferente que sigue codificando el mismo aminoácido [30]. Por ejemplo, los codones AAA y AAG codifican ambos el aminoácido lisina; en el caso de que hubiera un cambio AAG por AAA, no se produciría un cambio en la secuencia de la proteína codificada, por lo que no se espera que dichas la variante cambie la función de las proteínas codificadas (Figura 1.3). Por otro lado puede ocurrir que un cambio en un solo nucleótido provoque un cambio de aminoácido (cambio no sinónimo, *missense*). Por ejemplo, si mantenemos el ejemplo de los codones que codifican lisina, un cambio de AAG a AGG conducirá a la incorporación de arginina en la proteína resultante en lugar de ser lisina. Tales cambios podrían conducir a la pérdida completa de la función de la proteína resultante, o a un cambio drástico en la función, o podrían tener sólo un pequeño efecto que puede ser tolerado. Esto depende del contexto del aminoácido dentro de la proteína madura, de su función y de las características fisicoquímicas de los aminoácidos que se intercambian. En el ejemplo planteado anteriormente, la sustitución no sinónima produce un cambio de aminoácido que puede clasificarse como conservativo, o un cambio a un aminoácido que tiene propiedades fisicoquímicas similares al

[4] *Structural Variants*, en inglés.
[5] *Copy Number Variants*, en inglés.

aminoácido original (aminoácido básico por uno básico). Un cambio menos tolerado podría ser aquel no conservativo, en el que las propiedades del nuevo aminoácido generado sean distintas al original; en caso del ejemplo ya planteado, un cambio de codón AAG por ACG, produciría un cambio de lisina (básico) por trenonina (ácido). Los cambios no sinónimos no conservativos, a su vez pueden considerarse como semi-conservativos cuando, por ejemplo, se cambia un aminoácido cargado negativamente por otro cargado positivamente; o radical, cuando las propiedades son muy diferentes. Un tipo de cambio no sinónimo también pueden ser las variantes *nonsense*, las cuales provocan que el codón de un aminoácido se cambie por uno de los tres codones de parada (se produce la ganancia de un codón *stop*, por lo que se les denomina variantes *stopgain*). En contexto del ejemplo anterior, un cambio de AAG a TAG, generaría al cambio del codón para lisina por un codón de parada en el sitio de la variante. La traducción de la secuencia codificante mutada resultante conduce a la formación de un polipéptido truncado prematuramente, y la mayoría de estos truncamientos dan lugar a proteínas no funcionales (un caso excepción donde el cambio pueda tener menos efecto, puede ser que el mismo se produzca hacia el extremo C-terminal de la proteína) [30, 46]. Una variante *stopgain* puede darse por una SNV no sinónima, como la planteada en el ejemplo, pero también por una inserción/deleción con desplazamiento de marco de lectura, una inserción/deleción sin desplazamiento de marco de lectura o una sustitución en bloque. De la misma manera que un cambio puede producir una ganancia de un codón *stop*, se podría conducir a la eliminación de un codón de parada en el lugar de la variante o más adelante por el desplazamiento de marco de lectura (*stoploss*). Este tipo de variantes provocan que la traducción continua de una cadena de ARNm en lo que debería ser una región no traducida. En este caso la mayoría de los polipéptidos resultante de un gen con una variante *stoploss* pierden su función debido a su extrema longitud y al impacto en el plegamiento normal [47]. En la Figura 1.3 se muestra una representación esquemática de los ejemplos planteados anteriormente de cambios puntuales sobre un codón.

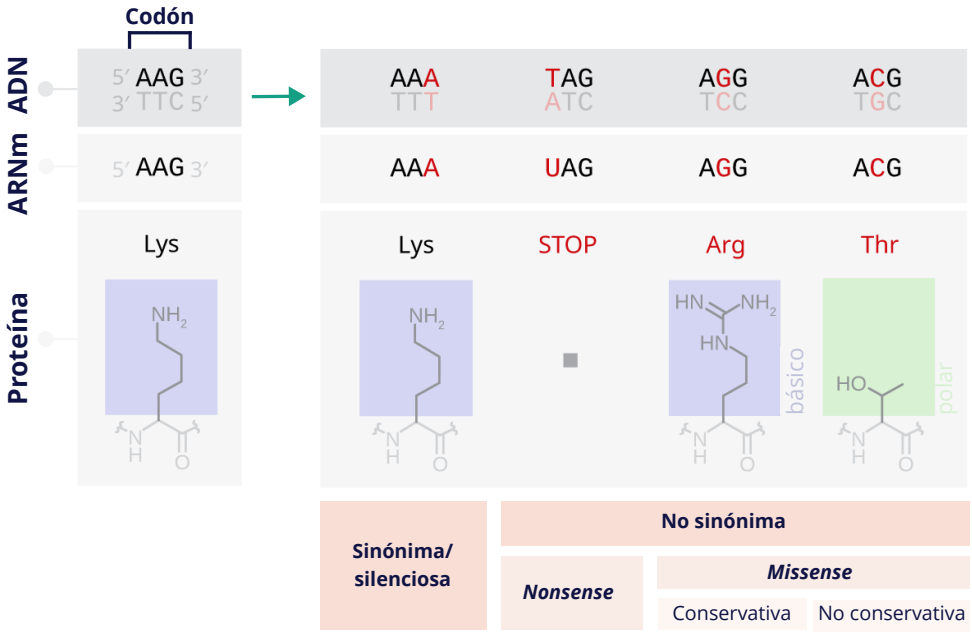


Figura 1.3: Ejemplo de efecto de SVN sobre un codón, clasificada por su impacto en la secuencia proteica. A la izquierda se muestra el codón en secuencia de ADN, ARNm y proteína normal. A la derecha se muestran las mismas secuencias, con cambios puntuales y su efecto funcional. Modificado de [48].

Anteriormente fueron descritos los efectos que pueden tener cambios puntuales en la secuencia de ADN. No obstante, los INDELs producen otros tipos de cambios en la secuencia, y su efecto

también va a depender de la región en la que se encuentren. Los INDELS que se producen en secuencias que controlan los niveles de expresión génica o el empalme de transcritos darán lugar a una expresión génica o empalme aberrantes, sin embargo el efecto de los INDELs en secuencias codificantes va a depender del número real de nucleótidos insertados o eliminados [30]. La deleción de tres (o un múltiplo de tres) nucleótidos de una secuencia codificante corresponde a la supresión de uno (o más) codones y dará lugar a la expresión de una proteína en la que se suprimen uno (o más) aminoácidos, sin cambios en la secuencia de aminoácidos restante. En caso de ser una inserción, sigue el mismo principio: a inserción de tres (o un múltiplo de tres) nucleótidos conduce a la inserción de uno (o más) aminoácidos en la proteína traducida. En estos casos nos encontramos ante lo que se conoce como *non frameshift deletion*, o *non frameshift insertion*, respectivamente. El nombre se debe a que los cambios no producen un desplazamiento del marco de lectura. Los polipéptidos que surgen como resultado, generalmente pueden seguir siendo funcionales [30]. Sin embargo, si la deleción o inserción se da en un número de nucleótidos no divisible por tres de una región no codificante, todos los codones subsiguientes se alterarán, desplazándose el marco de lectura. Esto da lugar a variantes *frameshift deletion* y *frameshift insertion*, respectivamente. Esto tiene su fundamento en que un ribosoma traduce las moléculas de ARNm de triplete en triplete. Si se suprime un solo nucleótido, el ribosoma seguirá traduciendo un triplete cada vez, pero a partir del punto de supresión o inserción, cada triplete diferirá de los presentes originalmente. Esto dará lugar a la formación de un polipéptido cuya secuencia diferirá completamente de la original, a partir del sitio donde se da el cambio [30]. Los desplazamientos del marco de lectura también pueden darse por sustituciones en bloque de uno o más nucleótido que causen cambios en a proteína codificada (*frameshift block substitution*). Con frecuencia, estos cambios de marco generan que se introduzcan codones de parada en el marco de lectura después del punto de deleción o inserción. En los casos en los que se eviten mecanismos como *nonsense-mediated decay* (NMD), la proteína puede resultar truncada [49, 50, 51]. En la Figura 1.4 se muestra el ejemplo de una *frameshift insertion* que produce la creación de un codón de parada luego del punto de inserción, generando un producto truncado [52].

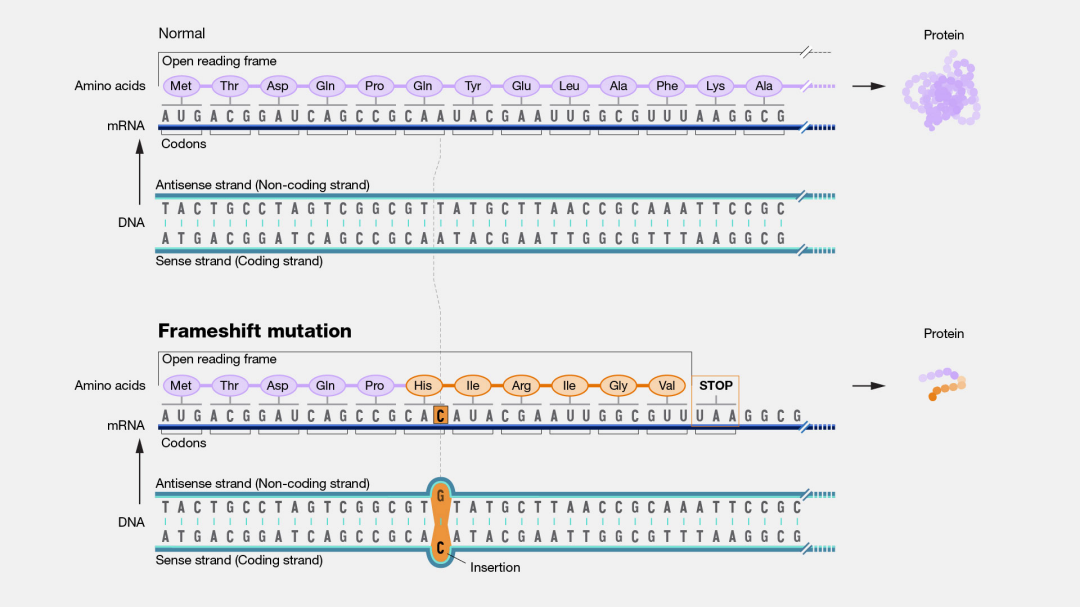


FIGURA 1.4: Ejemplo de una *frameshift insertion* que produce la creación de un codón de parada luego del punto de inserción. La proteína obtenida se encuentra truncada, dado el codón de parada prematuro. Extraído de [52].

Anteriormente fueron planteados efectos sobre variantes ubicadas en genes, en donde el efecto se da sobre la proteína que surge como producto, sin embargo los cambios también pueden ocurrir en

regiones no-codificantes del genoma. A pesar de que el efecto no sea directamente sobre una proteína, y que el entendimiento en general de las regiones no traducidas del genoma es muy reducido, se ha asociado el funcionamiento de ARN no codificante y pseudogenes con procesos que contribuyen a la regulación de la expresión génica [53, 54]. Es por ello que los cambios en regiones no codificantes pueden alterar la expresión de un gen, afectando de esta manera el proceso en el que actúa su producto [55].

1.2.5. Identificación e interpretación de variantes

La aplicación de tecnologías de NGS han brindado un gran avance en la identificación de la causa genética de enfermedades mendelianas raras. El rápido desarrollo tecnológico, así como la introducción de WES y los paneles de secuenciación de genes diana, facilitaron la aplicación de la NGS al análisis de la variación del genoma humano y al diagnóstico molecular de enfermedades hereditarias en general [56]. Desde la introducción de las tecnologías de WES y WGS en 2010, el ritmo de descubrimiento de genes subyacentes a ER por año ha aumentado. Entre 2012 y 2016 se identificaron más de 100 nuevas asociaciones enfermedad-gen por año en promedio [57, 58, 59]. El uso generalizado de las tecnologías de NGS permite detectar una gama completa de variantes genéticas comunes y raras de distintos tipos en casi todo el genoma, lo que facilita la investigación de enfermedades raras y las aplicaciones clínicas, y puede mejorar el descubrimiento de enfermedades comunes y la anotación de las variantes causales. En los últimos años se han desarrollado innovaciones tecnológicas y computacionales que han permitido incorporar tanto las metodologías de secuenciación del ADN como los algoritmos bioinformáticos a la rutina asistencial de la genómica médica de diagnóstico, identificando dichas variantes.
En la Figura 1.5 se muestra cómo es el proceso general computacional llevado a cabo para el descubrimiento de variantes y su genotipado a partir de la secuencia de ADN obtenida de tecnologías de NGS. Dado que este trabajo tiene un enfoque particular en variantes cortas de la línea germinal, los procesos presentados estarán asociados a la detección, procesamiento y análisis de dicho tipo de variantes específicamente.

El primer paso del proceso consiste en la secuenciación de los datos de las muestras, que en un contexto de genómica médica se trataría de los pacientes y/o sus familiares directos. Esto se puede realizar mediante técnicas de WES y/o WGS. La WGS implica la secuenciación de todo el genoma de un individuo, incluidas las regiones codificantes y no codificantes de la secuencia de ADN. Este enfoque proporciona una visión completa de la composición genética de un individuo y permite la identificación de variantes en cualquier región del genoma. El proceso de WGS implica la fragmentación del ADN en fragmentos más pequeños, seguida de la secuenciación de cada fragmento y el posterior ensamblaje de la secuencia genómica completa [59]. La WES, por su parte, se centra específicamente en la secuenciación del exoma o las regiones codificantes del genoma. Este enfoque puede ser más rentable que la WGS y puede resultar especialmente útil en los casos en los que se sospecha que la base genética de una enfermedad se encuentra en las regiones codificantes de proteínas. La WES implica, antes de la secuenciación, la captura de las regiones específicas del exoma [59].
Los datos generados a través de la secuenciación y el análisis son de una enorme cantidad (alrededor de cientos de gigabytes a terabytes). Para analizar la cantidad masiva de datos, se han desarrollado distintas herramientas y *pipelines*[60] para el abordaje computacional de estos datos. En este sentido, entre los flujos de trabajo más desarrollados y utilizados se encuentran las "*best practices*" de GATK, que proporcionan recomendaciones paso a paso para realizar análisis de detección de variantes en datos de secuenciación [61]. Existen varios *pipelines* adaptados a aplicaciones concretas en función del tipo de variación de interés y de la tecnología empleada. En particular se describirán a modo general un *pipeline* típico de detección de variantes cortas en la línea germinal a partir de datos de WGS o WES.
Los secuenciadores proporcionan lecturas formato FASTQ, que contiene los *reads* secuenciados. Luego

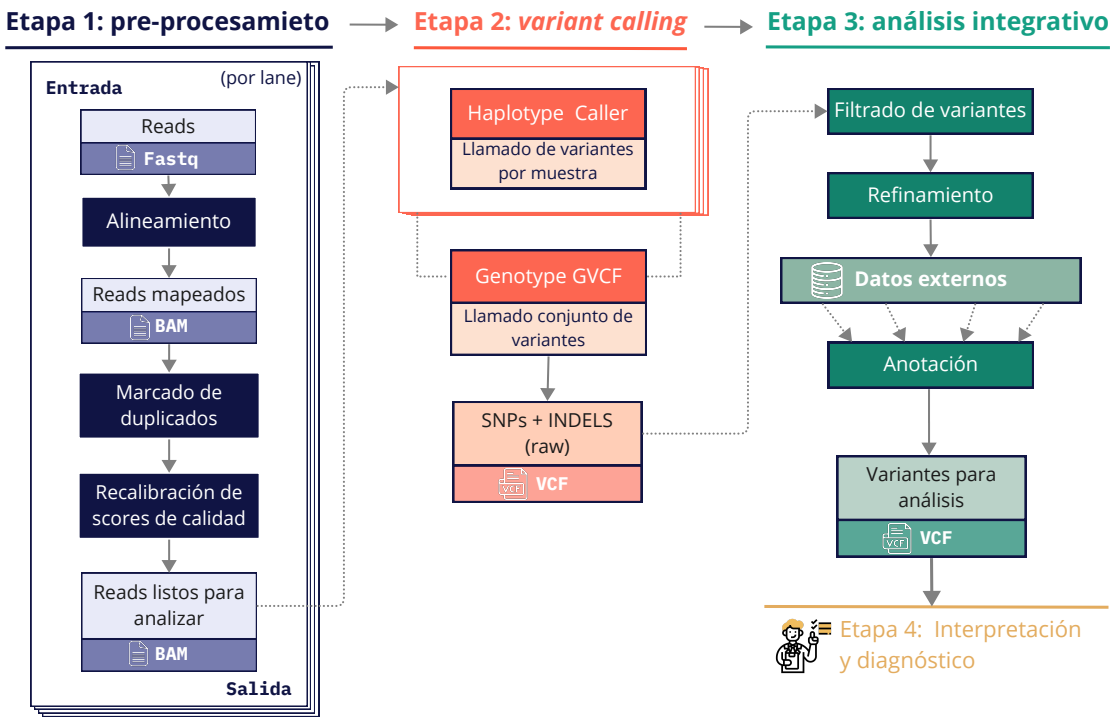


FIGURA 1.5: *Pipeline* de procesamiento de datos de secuenciación (WGS o WES) para la detección de variantes cortas en la línea germinal (SNPs e INDELs). El proceso de basa en las *best practices* de GATK, y parte de datos crudos de secuenciación hasta la adquisición de un conjunto de variantes prontas para su interpretación clínica. Los tres grandes pasos previos a la interpretación consisten en: el pre-procesamiento de datos de secuenciación, *variant calling*, y post-procesamiento de las variantes adquiridas.

de pre-procesar los *reads* realizando controles de calidad, la corrección de errores de secuenciación, eliminación de adaptadores, entre otros pasos, se comienza el pipeline, que en líneas generales, consiste en los pasos a continuación. Solo serán desarrollados aquellos puntos más relevantes o explicativos en el contexto del trabajo realizado:

- Pre-procesamiento:

  1. Alineamiento: consiste en mapear los *reads* contra un genoma de referencia. La mayoría de los procedimientos modernos utilizan BWA-MEM para dicho alineamiento [62, 63], a través del cual se obtiene un archivo SAM/BAM con los *reads* alineados [64]. El genoma de referencia con el que se compara es una construcción o ensamblaje que representa un consenso de las secuencias de ADN de varios individuos, con el objetivo de representar la versión más común del genoma humano [65, 66]. El genoma humano de referencia ha pasado por varias versiones desde su publicación inicial en 2001, siendo las más recientes la GRCh38 (también conocida como hg38) y su predecesora, la GRCh37 (también conocida como hg37 o hg19). GRCh38/hg38 es el ensamblaje del genoma humano publicado en diciembre de 2013, que utiliza *contigs* alternativos o ALT para representar la variación compleja común y corrige artefactos de secuenciación que causan falsos SNPs e INDELs cuando se utiliza el ensamblaje GRCh37 (b37/Hg19) [67]. Más allá de las mejoras implementadas en la referencia GRCh38/hg38, en el trabajo actual se trabajará con datos alineados contra la referencia GRCh37/hg37, dada la cantidad de insumos generados para dicha referencia, al momento de desarrollar el proyecto.

2. Marcado de duplicados: consiste en la identificación y el etiquetado de lecturas duplicadas en un archivo SAM/BAM, entendiendo como lecturas duplicadas a aquellas originadas a partir de un único fragmento de ADN. Estas lecturas duplicadas pueden surgir durante la preparación de la muestra, por ejemplo, en la construcción de bibliotecas mediante PCR[6], y la identificación de variantes en las mismas, puede resultar en estadísticas erróneas respecto a la variante en caso de que no sean tenidas en cuenta.
3. Recalibración de *scores* de calidad de bases: consiste en la detección de errores sistemáticos cometidos por el secuenciador al estimar la precisión de cada llamada de base. Los *scores* de calidad generados por los secuenciadores están sujetos a diversas fuentes de error, lo que puede generar *scores* de calidad sobreestimados o subestimados. Mediante la recalibración se modelan empíricamente estos errores y posteriormente se ajustan los *scores* de calidad.

- *Variant Calling* (detección de variantes):

1. *Haplotype Caller*: este proceso consiste en la determinación de SNPs e INDELs mediante el reensamblaje local del haplotipo (cuando se encuentra una región con signos de variación, se descarta la información de mapeo existente y vuelve a ensamblar completamente las lecturas de dicha región) [68]. En este proceso, una de las herramientas más utilizadas, y utilizada por el grupo de trabajo es GATK [69, 70]. En este proceso se parte de un archivo BAM y se obtiene un conjunto de variantes crudas mediante un archivo VCF: el formato estándar para el almacenamiento de variantes genómicas [71] (ver Apéndice B).
2. *Recalibración o filtrado*: se construye un modelo de recalibración para puntuar la calidad de las variantes con fines de filtrado, y se aplican filtros de calidad a cada variante, los cuales se asignan al campo `INFO` del archivo VCF (ver estructura en Apéndice B).

- *Anotación*: se asigna información a cada variante de diversas fuentes (este proceso será desarrollado más adelante).

Una vez que las variantes son anotadas, se pueden aplicar distintos tipos de filtros a los efectos de obtener un sub-conjunto de variantes de interés clínico. Este sub-conjunto es posteriormente analizado, en un contexto de genómica médica, para la realización de un diagnóstico. Aquí es donde el rol del intercambio entre la bioinformática y la práctica clínica es fundamental. A partir de aquí, comienza una de las tareas más complejas del proceso: la interpretación de las variantes. El trabajo actual se da en el contexto de la tercera etapa del procesamiento, contemplando los procesos posteriores a la obtención de las variantes, sin embargo, la comprensión de los procesos que subyacen a todo el *pipeline* es fundamental para un desarrollo acorde.

1.2.5.1. Interpretación y clasificación de variantes

La información derivada de la clasificación de variantes es fundamental para descubrir o confirmar las etiologías de las enfermedades y orientar las pautas de tratamiento y los planes específicos para cada paciente. Sin embargo, la interpretación de variantes es una de las tareas más complejas del proceso. La interpretación clínica de los datos de secuenciación requiere tanto la estandarización de las pautas de clasificación de variantes como la coherencia en el flujo de trabajo y las pruebas consideradas a la hora de determinar la relación entre una variante y el fenotipo de una enfermedad. En 2015, el American College of Medical Genetics and Genómica (ACMG) y la Asociación de Patología Molecular (AMP) publicaron en conjunto una serie de lineamientos para la clasificación de variantes con el fin de homogeneizar los métodos y criterios y reducir la discordancia entre los laboratorios clínicos, genetistas clínicos y otros profesionales sanitarios implicados en la interpretación de variantes observadas en pacientes con presuntos trastornos hereditarios, principalmente mendelianos, en un

[6] *Polymerase Chain Reaction*, en inglés-

contexto clínico [44]. Las reglas recomiendan clasificar las variantes en cinco categorías discretas, en función de la fuerza de las pruebas que apoyan su patogenicidad:

- Patogénicas (P): Existen pruebas sólidas de que la variante causa determinado trastorno.
- Probablemente patogénica (LP[7]): Existen pruebas moderadas de que la variante es causante de determinado trastorno.
- Variantes de significado incierto (VUS[8]): No hay pruebas suficientes para clasificar la variante como patogogénica o benigna.
- Probablemente benigna (LB[9]): existen pruebas moderadas de que la variante no es causante de enfermedad.
- Benigna (B): existen pruebas sólidas de que la variante no es causante de enfermedad.

Para asignar alguna de las categorías mencionadas anteriormente, se plantea el uso de 28 criterios agrupados en un conjunto para la clasificación de variantes patogénicas o probablemente patogénicas y otro para la clasificación de variantes benignas o probablemente benignas. Los criterios se dividen en 16 criterios patogénicos y 12 criterios benignos, e incluyen una medida relativa de fuerza para cada evidencia a favor o en contra de la patogenicidad: independiente, muy fuerte, fuerte, moderada o de apoyo. Según según esta medida relativa de fuerza, los criterios patogénicos se dividen en muy fuertes (PVS1), fuertes (PS1-4), moderados (PM1-6) o de apoyo (PP1-5). Por otro lado, los criterios benignos se dividen en independientes (BA1), fuertes (BS1-4) o de apoyo (BP1-7). Todos los criterios de una misma categoría tienen la misma ponderación (la numeración no tiene implicancias de peso, sino de nomenclatura). Como puede observarse, la primera letra de cada código con los que se representa la evidencia indica si apoyan una clasificación patogénica (P) o benigna (B); la segunda letra indica el peso: VS muy fuerte (*very strong*), S fuerte (*strong*), M moderada (*moderate*), P de apoyo (*supporting*), A independiente (*stand-alone*). En las Tablas A.1 y A.2 se presenta un detalle de cada criterio, para clasificaciones patogénicas y benignas, respectivamente.
Una vez que se establecen los criterios que se cumplen para la variante en evaluación, se aplican reglas de combinación de criterios (Tabla A.3), que derivan en la clasificación de 5 categorías descrita anteriormente, atendiendo a que las variantes se clasifican como VUS cuando no se cumplen los criterios planteados, o si son contradictorios [44].
Los criterios sugeridos también pueden agruparse según el tipo o fuente de evidencia que requieren (Figura 1.6) [72]:

- Datos poblacionales: los datos de frecuencia alélica menor (MAF[10]) a nivel poblacional son fundamentales, dado que es esperado que los alelos causantes de enfermedades para la mayoría de los trastornos mendelianos sean raros (con baja frecuencia). Hay cinco criterios (BA1, BS1, BS2, PM2 y PS4) que utilizan estos datos al momento de acumular evidencia. Por ejemplo, una MAF mayor a 5 % en cualquier población global se considera una clasificación benigna "independiente" (BA1) para la gran mayoría de los trastornos mendelianos, con la excepción de alelos fundadores bien conocidos. La evaluación de estos criterios se logra buscando la variante en cuestión en bases de datos de población disponibles públicamente, tales como 1000 Genomes [73], Exome Sequencing Project (ESP) [74], Exome Variant Server [75], Exome Aggregation Consortium (ExAC) [76].

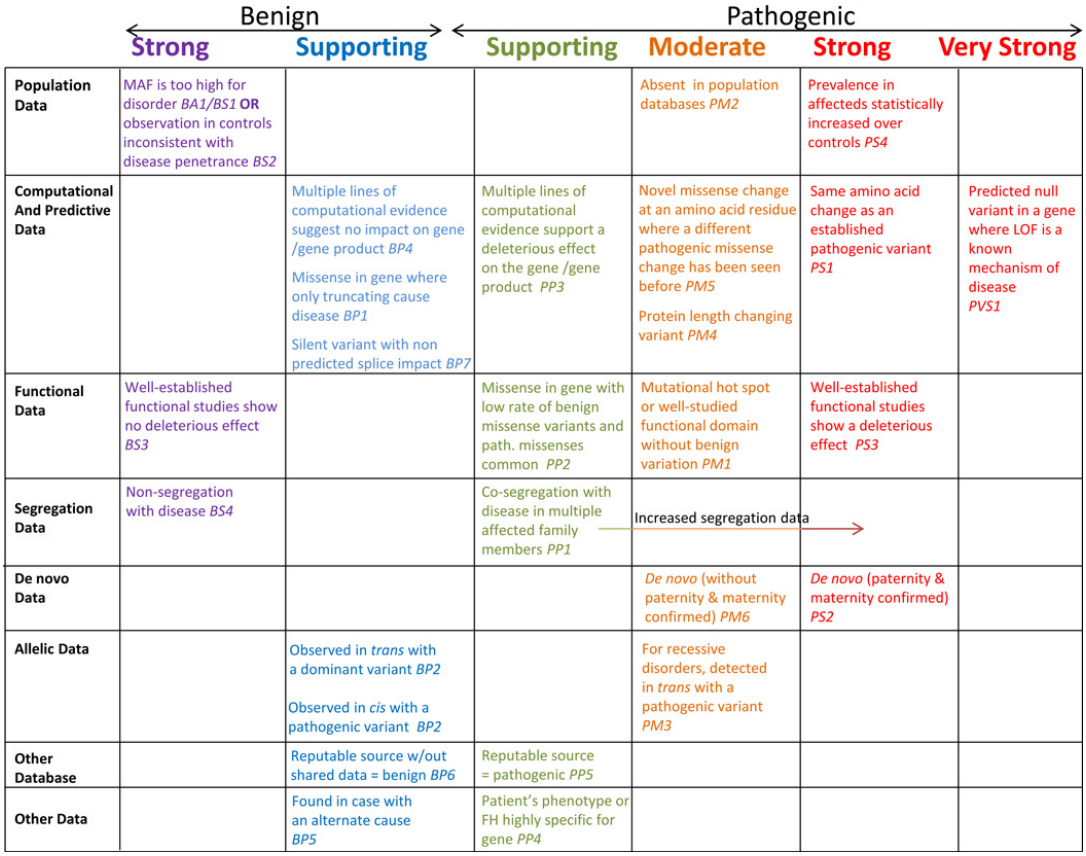
[7] *Likely Pathogenic*, en inglés
[8] *Variant of Uncertain Significance*, en inglés
[9] *Likely Benign*, en inglés
[10] *Minor Allele Frequency*, en inglés.

- Evidencia alélica y co-segregación: debido a los patrones mendelianos de herencia que se observan en la mayoría de los trastornos monogénicos, la evidencia de segregación en los miembros de la familia (o la falta de ella) puede informar sobre la interpretación de las variantes. En este sentido, hay 4 criterios a los que se aplican evidencias de segregación. (PS2, PM6, PP1 y BS4). Por ejemplo, la aparición de novo de una variante se considera una prueba contundente de patogenicidad (PS2) cuando se confirman la maternidad y la paternidad, la variante se encuentra en un gen asociado a una enfermedad consistente con el fenotipo del paciente, y no hay antecedentes familiares de enfermedad. Aparte de la aparición de novo y del análisis de cosegregación, los estudios familiares también pueden utilizarse para determinar la fase de dos o más variantes heterocigotas, lo que se aplica a tres criterios (PM3, BP2 y BP5). Por ejemplo, si una variante se produce en un gen asociado a una afección autosómica recesiva y está en trans con una variante patogénica, puede aplicarse el criterio PM3.

- Datos computacionales y predictivos: estos criterios están relacionados con el tipo de variante en cuestión y su impacto previsto en el producto proteico basado en el conocimiento de la función, estructura y conservación evolutiva de la proteína, e incluyen el uso de predictores *in sílico*. Hay 9 criterios que se basan en este tipo de evidencia (PVS1, PS1, PM1, PM4, PM5, PP2, PP3, BP1, BP3, BP4 y BP7). En conjunto, estos criterios proporcionan una forma de predecir la patogenicidad de las variantes extrapolando lo que ya se sabe sobre el impacto funcional y clínico de variantes similares. En virtud de la naturaleza predictiva de esta categoría, estos criterios se aplican más adecuadamente a variantes en genes con un mecanismo molecular y dominios funcionales bien comprendidos [72].

- Datos funcionales: datos de ensayos funcionales bien establecidos son una herramienta poderosa en apoyo de la patogenicidad. En este caso, hay dos criterios que se basan en los mismos: los que muestran un efecto deletéreo (PS3) o ningún efecto (BS3).

- Otros criterios: Los criterios restantes (PP4, PP5 y BP6) pueden aplicarse a distintos tipos. PP4 es un caso donde se aplica el fenotipo observado en apoyo de de una clasificación. Si bien en general el hecho de que un paciente presente un fenotipo que coincida con el espectro conocido de características clínicas de un gen no se considera una prueba de patogenicidad, el fenotipo del paciente puede considerarse una prueba de apoyo en determinadas condiciones (por ejemplo, si el paciente tiene una historia familiar coherente con el modo de herencia, las pruebas son sensibles y el gen tiene poca variación benigna asociada) [44]. En el caso de PP5 y BP6, se basan en tomar como fuente otros laboratorio con una amplia experiencia en el área, cuyo reporte se haya compartido en bases de datos, para determinar el significado clínico de la variante.



	Benign		Pathogenic			
	Strong	Supporting	Supporting	Moderate	Strong	Very Strong
Population Data	MAF is too high for disorder *BA1/BS1* **OR** observation in controls inconsistent with disease penetrance *BS2*			Absent in population databases *PM2*	Prevalence in affecteds statistically increased over controls *PS4*	
Computational And Predictive Data		Multiple lines of computational evidence suggest no impact on gene /gene product *BP4* Missense in gene where only truncating cause disease *BP1* Silent variant with non predicted splice impact *BP7*	Multiple lines of computational evidence support a deleterious effect on the gene /gene product *PP3*	Novel missense change at an amino acid residue where a different pathogenic missense change has been seen before *PM5* Protein length changing variant *PM4*	Same amino acid change as an established pathogenic variant *PS1*	Predicted null variant in a gene where LOF is a known mechanism of disease *PVS1*
Functional Data	Well-established functional studies show no deleterious effect *BS3*		Missense in gene with low rate of benign missense variants and path. missenses common *PP2*	Mutational hot spot or well-studied functional domain without benign variation *PM1*	Well-established functional studies show a deleterious effect *PS3*	
Segregation Data	Non-segregation with disease *BS4*		Co-segregation with disease in multiple affected family members *PP1*	Increased segregation data →		
De novo Data				*De novo* (without paternity & maternity confirmed) *PM6*	*De novo* (paternity & maternity confirmed) *PS2*	
Allelic Data		Observed in *trans* with a dominant variant *BP2* Observed in *cis* with a pathogenic variant *BP2*		For recessive disorders, detected in *trans* with a pathogenic variant *PM3*		
Other Database		Reputable source w/out shared data = benign *BP6*	Reputable source = pathogenic *PP5*			
Other Data		Found in case with an alternate cause *BP5*	Patient's phenotype or FH highly specific for gene *PP4*			

FIGURA 1.6: Organización de los criterios ACMG/AMP por tipo de evidencia y peso de los criterios para una clasificación benigna (lado izquierdo) o patogénica (lado derecho). Extraído de [44].

En la Figura 1.6 se muestra cómo se organiza cada uno de los criterios según el tipo de evidencia, así como el peso de los criterios para una clasificación benigna (lado izquierdo) o patogénica (lado derecho) [44].

1.3. Objetivos

1.3.1. Objetivo general

En el presente trabajo se buscó un primer acercamiento en el desarrollo/implementación de herramientas de clasificación de variantes genómicas, basado en enfoques se aprendizaje automático. Dichas herramientas deberían contemplar información a partir de diversidad bases de datos públicas, además de insumos provenientes de expertos en el área. La obtención de estos últimos debería generarse a través de una plataforma de intercambio y aprendizaje, la cual contemple de forma objetiva el aporte humano. Se buscó que tanto la obtención de datos como el desarrollo de los algoritmos hagan énfasis en la clasificación de variantes potenciales causantes de enfermedades raras.

1.3.2. Objetivos específicos

- Implementación de estrategias que permitan la clasificación de variantes con un significado incierto e interpretaciones conflictivas de patogenicidad, con un énfasis particular en desambiguar las interpretaciones conflictivas.
- Diseño, implementación, testeo y aplicación inicial algoritmo de clasificación de variantes cortas basado en aprendizaje automático.

- Recolección de datos genómicos a partir de bases de datos públicas para el entrenamiento de los algoritmos.
- Evaluación de la metodología existente, buscando la que mejor se adapte al problema y los requerimientos.
- Desarrollo de una plataforma de prestaciones básicas para obtención etiquetas emitida por usuarios que se desempeñen en el área de la genómica médica.
- La plataforma desarrollada debe contemplar los diferentes niveles de pericia de los usuarios, además de fomentar formas de generar aprendizaje en la clasificación o priorización de variantes.
- Desarrollo/implementación de herramientas con potencialidad en la identificación de variantes variantes cortas en la línea germinal asociadas a enfermedades raras.

1.4. Esquema de trabajo general

Para abordar la problemática y los objetivos propuestos se trabajó en el desarrollo de dos grandes componentes de un posible sistema global: una plataforma en sus instancias preliminares de recolección de datos y aprendizaje para expertos y/o aprendices en el área de genómica médica, y un sistema de priorización de variantes basado en enfoques de aprendizaje automático. La relación y el esquema de la implementación general pueden observarse en la Figura 1.7.
La plataforma desarrollada consiste en una aplicación web de prestaciones básicas que nuclea dos funcionalidades principales: el etiquetado de variantes por usuarios (que aporta insumos al conjunto de datos del sistema de clasificación), y el entrenamiento de usuarios en la tarea de clasificación de variantes.
Los potenciales clientes de la plataforma serán médicos, genetistas, profesionales clínicos, bioinformáticos, y en general personal que se desempeñe en el área de la genómica médica y que se enfrente a la clasificación o determinación de patogenicidad de variantes como tarea habitual. Esta tarea, la cual es crucial al momento de buscar el diagnóstico mediante el uso de herramientas NGS, requiere de práctica y de la generación de una metodología de evaluación de las variantes a estudiar. La funcionalidad de entrenamiento de usuarios busca instrumentar a los mismos con las herramientas necesarias para lograr dicha metodología y práctica, estableciendo los estándares básicos necesarios para la tarea. De esta forma, se buscó que la plataforma sea un espacio donde los usuarios puedan clasificar variantes a modo de entrenamiento, basándose en un sistema de niveles de pericia y medios para avanzar de niveles. De esta forma, se sentarían las bases de una plataforma que mediada por un problema a modo de juego, ayude a los expertos a mejorar sus habilidades de clasificación. A medida que los niveles crecen, crece la dificultad de las variantes a clarificar, hasta que el último nivel consiste en aportar al sistema de clasificación de variantes mediante aprendizaje automático, a través del etiquetado.
El etiquetado de variantes consiste en un proceso similar al de entrenamiento en cuanto a la información brindada al usuario a través de la plataforma web, con excepción de que solamente se espera que el mismo agregue la etiqueta correspondiente al nivel de patogenicidad de la variante: “benigna”, “probablemente benigna”,“patogénicas”, “probablemente patogénica”, y “variante de significado incierto” (inicialmente). Esta última funcionalidad, contará con una base de variantes las cuales contienen conflictos en su clasificación, o sobre las cuales no se generó un consenso. Es así como las variantes luego pasan por un proceso de clasificación que contemplará el nivel de los usuarios en su aporte.
Las etiquetas obtenidas serán destinadas al clasificador de variantes, el cual usa un enfoque de aprendizaje supervisado para asignar las variantes estudiadas a 4 categorías idealmente: Benigna, Probablemente benigna, Patogénica y Probablemente patogénica. Este clasificador está basado en un rico conjunto de características genómicas y fenotípicas, entre las cuales hay características relacionadas

con la frecuencia alélica de la población, características basadas en las directrices ACMG/AMP, *scores* de predicción *in silico* de patogenicidad existentes, características de impacto funcional de las variantes y características a nivel de genes.
La implementación estará enfocada principalmente a variantes cortas de la línea germinal, con la posibilidad de aplicación a otros tipos de variantes en implementaciones futuras.

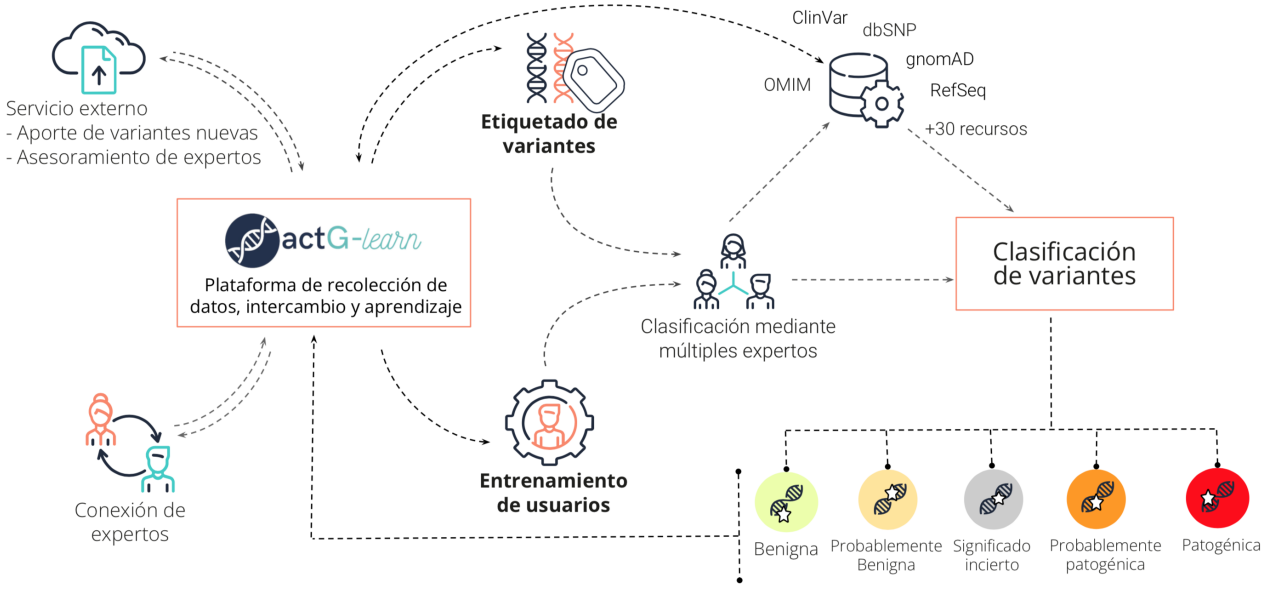


FIGURA 1.7: Esquema general del trabajo realizado, el cual consiste en dos partes, que buscan aportar al mismo cometido: la clasificación de variantes cortas en el genoma humano. Por un lado, el sistema tiene una plataforma web con dos funcionalidades: la recolección de etiquetas brindadas usuarios luego de la evaluación de variantes, y el entrenamiento de usuarios en la tarea de clasificación. Esta plataforma se nutre de las clasificaciones realizadas por distintos usuarios, y de múltiples recursos y bases de datos públicas. Las etiquetas brindadas, las cuales generan un consenso que contempla el nivel de pericia de cada usuario, aportarían en un futuro a un sistema de clasificación basada en herramientas de aprendizaje automático, con el objetivo de asignar etiquetas a variantes sin un significado asociado, o con etiquetas conflictivas.

1.5. Estructura de la documentación

El presente trabajo está dividido en cinco grandes partes: capítulo introductorio (actual), antecedentes, materiales y métodos comunes, plataforma de aprendizaje, clasificación de variantes, y conclusiones finales.
En el capítulo introductorio se buscó dar un contexto al trabajo, sus fundamentos, plantear los objetivos del mismo, y la motivación detrás de todo el proyecto.
El segundo capítulo se centra en evaluar el estado del arte, los antecedentes y las herramientas similares existentes al momento y sus características.
El tercer capítulo abarca la descripción de los datos manipulados a lo largo de todo el trabajo, sus características y el abordaje mediante el cual se los procesa.
El cuarto capítulo se describe la plataforma de recolección, y aprendizaje generada, las decisiones de diseño detrás de la misma y las consideraciones respecto a su alcance, cómo se eligieron y usaron las herramientas para su desarrollo, la arquitectura general de la misma y su funcionalidad e implementación.
En el capítulo cinco se desarrolla el trabajo realizado en lo que respecta al uso de herramientas de aprendizaje automático para la clasificación o priorización de variantes genómicas.

Finalmente, se presentan conclusiones y comentarios finales.

Dado que el trabajo cuenta con dos grandes partes: la plataforma de aprendizaje y la clasificación de variantes se destinarán dos grandes capítulos auto-contenidos que describen cada una de estas partes. Internamente, cada capítulo tendrá su propio planteo inicial, metodología y/o implementación, resultados preliminares, discusiones, conclusiones y trabajo a futuro. Externamente a estos capítulos, se brindarán los aspectos que ambos tienen en común: inicialmente los fundamentos, una descripción de los datos usados en ambas partes y el procesamiento de los mismos, y finalmente las conclusiones del trabajo general.

Capítulo 2

Antecedentes

La clasificación de las variantes genómicas es un campo complejo y en constante evolución, y el uso de directrices y bases de datos estandarizadas es fundamental para una interpretación precisa y la toma de decisiones clínicas. Las reglas ACMG/AMP y otros sistemas de clasificación lograron proporcionar un marco para evaluar las pruebas de patogenicidad de las variantes y ayudan a garantizar la coherencia y la precisión en la interpretación de las variantes. A pesar de lo específicas que pueden ser las reglas, a menudo los laboratorios u organizaciones discrepan sobre el estado de patogenicidad de una variante. Para evidenciar esto, en 2016 se publicó un trabajo en el que el programa Clinical Sequencing Exploratory Research (CSER) [77] puso a prueba un conjunto de 99 variantes en nueve laboratorios de diagnóstico molecular y halló una concordancia aceptable dentro de un mismo laboratorio (79 %), pero escasa concordancia entre laboratorios (34 %). Incluso después de las discusiones entre laboratorios, la concordancia mejoró hasta sólo el 71 %. Este trabajo identificó varios ámbitos en los que las directrices carecían de especificidad o estaban sujetas a interpretaciones ambiguas o contradictorias [6]. A partir de allí se comenzó a trabajar en diversos refinamientos e implementaciones de estas reglas. Un ejemplo de refinamiento es "Sherlock" [78], que surgió en 2017 como una implementación de refinamientos de las reglas propuestas ACMG/AMP, en donde se llevaron a cabo 108 refinamientos bien detallados a estas reglas. Esto se logró mediante un proceso iterativo a través del cual se identificaron más de 40000 variantes asociadas a más de 500 genes, soportando un enfoque que se consideró más adecuado a la clasificación [78]. Un segundo ejemplo de esfuerzo de mejora es el método de puntuación iterativo basado en las reglas que proporciona una puntuación de riesgo de patogenicidad de variantes basada en información de grado clínico y ofrece un sistema de interpretación de variantes más estable [79]. Entre los aportes más grandes se encuentran los refinamientos aportados por ClinGen (Clinical Genome resource) [80], creado en el 2013 por el National Institutes of Health (NIH) en colaboración con ClinVar, como un recurso central autorizado para definir la relevancia clínica de las variantes para su uso en medicina de precisión e investigación. ClinGen ha trabajado perfeccionado y normalizado los criterios de evaluación, aportando una serie de lineamientos ampliamente usados en la actualidad [80, 81, 82, 83, 84]. Además de los refinamientos orientados a disminuir las discordancias e interpretaciones contradictorias de los evaluadores, habían otros aspectos que las reglas por si solas no cumplían: algoritmos específicos para aplicarlas teniendo en cuenta qué bases de datos aplicar, además de herramientas que combinen y concentren la enorme cantidad de bases de datos que están implicadas en el uso de las reglas. Para hacer frente a estos se hizo necesario el desarrollo de herramientas computacionales y servicios web fáciles de usar y automatizados que puedan generar criterios versionados y reproducibles para cada variante y ayudar a los intérpretes humanos a comprender el significado clínico de las variantes genéticas. En este estudio, una de las herramientas generadas fue InterVar, que partiendo de un VCF como entrada puede generar una interpretación automatizada de las reglas ACMG/AMP. Además, Intervar cuenta con un servidor web complementario, wInterVar [85], para facilitar la interpretación de variantes con pasos de interpretación automatizados y pasos de ajuste manual [86]. Otros ejemplos de sistemas de asistencia a los profesionales en la clasificación son aquellos desarrollados por ClinGen. Entre ellos, en 2017 se desarrolló una "calculadora de patogenicidad" basada en las reglas ACMG/AMP, que consiste en un sistema y servicio web que asiste en la evaluación de la patogenicidad de variantes de

origen germinal con herencia mendeliana[87]. Esta herramienta permitió reducir la aplicación manual de las reglas, buscando atacar el problema de ser una tarea compleja y propensa a error humano [87]. Aquí el usuario puede aplicar etiquetas de kas reglas a un alelo específico, asociándolo a datos que soportan la generación de una etiqueta de patogenicidad de forma guiada. Por otro lado, el trabajo más actual es "*ClinGen Variant Curation Interface*" (VCI), una plataforma web de interpretación de variantes en la línea germinal que que también ayuda al usuario en la aplicación de criterios de evidencia y clasificación de variantes basados basados en reglas ACMG/AMP [88]. En este caso la plataforma permite la colaboración y la revisión por pares entre los paneles de expertos curadores de ClinGen, ayudando a los usuarios a identificar, anotar y compartir la evidencia para clasificar una variante. Los flujos de trabajo de navegación (basados en procesos de curación de variantes de la FDA) ayudan a los usuarios proporcionándoles orientación para aplicar los criterios de evidencia ACMG/AMP y documentar la procedencia para afirmar clasificaciones de variantes [88]. De esta forma la VCI ofrece una estrategia centralizada para la clasificación de variantes clínicas con un aporte grande en el aprendizaje desde el personal sanitario, facilitando la adopción estándares en la clasificación [88].

VarSome también surge como una plataforma que centraliza la información de más de 100 recursos y/o bases de datos externas, incluidas bases de datos públicas, literatura científica y anotaciones clínicas, para proporcionar un análisis exhaustivo de las variantes genéticas [89]. Allí los usuarios pueden buscar una variante específica, o ingresar variantes nuevas, para su evaluación. El sistema no solo presenta toda la información asociada a dicha variante, sino que también permite acceder a un reporte de la patogenicidad mediante su propio clasificador automático que implementa 21 de las reglas ACMG/AMP. En la versión más actual de la plataforma explica cada regla ACMG, junto con el motivo por el que se ha activado, o por qué no, y si el usuario cuenta con evidencia adicional a la que ya se encuentra allí, puede activar manualmente otras reglas y llegar al veredicto final de forma automática [90]. De esta forma, VarSome ofrece una serie de herramientas para el filtrado, la priorización y la interpretación de variantes, lo que la convierte en un valioso recurso para médicos, investigadores, asesores genéticos, entre otros. VarSome también genera una comunidad de evaluadores, en la que la clasificación de variantes se puede abordar de forma colaborativa, generando no solo un entorno para el intercambio, sino para generar una fuente valiosa de datos para su uso [89].

En este mismo camino se desarrolla Franklin, una plataforma que al igual que VarSome, proporciona la integración de múltiples fuentes de datos y flujos de trabajo para el análisis de variantes y datos de secuenciación masiva [91]. Franklin ofrece una serie de herramientas y funciones para la gestión de datos, el control de calidad, la identificación de variantes, la anotación y la interpretación. La plataforma que ofrece está diseñada para ser escalable y flexible, por lo que es adecuada para una amplia gama de aplicaciones, desde pequeños estudios hasta proyectos de secuenciación clínica a gran escala. Al igual que VarSome, Franklin representa un punto de conexión en el ámbito de la genómica médica, donde se tienden redes a los efectos de ampliar la información genómica disponible [91], esto es realizado gracias al desarrollo de una plataforma basada en aprendizaje automático específicamente para el intercambio.

El proceso de identificación de variantes deletéreas o contribuyentes al desarrollo de enfermedades entre las millones de variantes genéticas obtenidas a partir del genoma de un individuo, suele implicar una serie de pasos que son fuertemente dependientes del uso de herramientas bioinformáticas (anotación de variantes, filtrado de variantes, predicción *in silico*, interpretación clínica por parte de expertos humanos, etc.) [86, 92]. Se han desarrollado varias herramientas y bases de datos para ayudar a quienes se desempeñan en el área a comprender el impacto funcional de las variantes sobre los genes y el desarrollo de enfermedades. En general, estas herramientas pueden dividirse en distintas categorías, entre las que se evaluarán: herramientas de anotación, herramientas de predicción y bases de datos. A partir de los millones de variantes que se obtienen luego de un *variant calling*, es muy difícil comenzar a buscar variantes de interés. Para ello se requiere en general de una "ayuda" respecto a la información insuficiente (desde el punto de vista del interpretador) para comenzar a generar conjuntos de datos más reducidos y orientados. Por ejemplo, más allá de que se conoce la

ubicación de una variante a través de los campos comunes en un VCF, es de ayuda conocer para cada variante la región a la que esa posición corresponde en el genoma, y qué efecto genera dicha variante. Para ello es que, como fue visto en la introducción, se hace un proceso final de anotación, previo a la interpretación, mediante herramientas específicas. Las herramientas de anotación son mecanismos para predecir el efecto de las variantes y/o aportar la información disponible de las mismas proveniente de otras fuentes, recursos o bases de datos. Hay disponibles una serie de herramientas de anotación, como ANNOVAR [93] (sobre la cual se detallará más adelante), SNPeff [94], VEP [95], VAAST [96], SeattleSeq [97], y herramientas más actuales como My Variant Info [98], pueden predecir cómo afectan las variantes a la estructura de los transcritos o a las secuencias codificantes. Por ejemplo, una tarea común de estas herramientas es clasificar las variantes en intrónicas, intergénicas, de *splicing* y exónicas, y para las variantes exónicas, pueden calcular cómo se ven afectadas las secuencias de aminoácidos. Este tipo de herramientas, si bien usan información genómica exclusivamente, fue mejorado posteriormente por la integración de fenotipos de pacientes en los análisis, por ejemplo en herramientas como Phevor [99], Exomiser [100], Phenolyzer [101], y más recientemente AMELIE [102]. En segundo lugar, hay diversas herramientas pueden predecir si la variante es perjudicial para la función o la estructura de la proteína utilizando distintos tipos de información, basadas en la información disponible y en la amplia potencia computacional actual. Estos, como fue visto anteriormente, permiten ser aplicados en varias de las reglas ACMG/AMP. Más adelante, se detallará sobre algunos de ellos en específico. Por otro lado, las bases de datos públicas específicas de enfermedades y genes, como la Human Gene Mutation Database (HGMD) [103], ClinVar [104] y varias bases de datos específicas de locus, pueden documentar variantes genéticas validadas funcional o clínicamente que son patogénicas para enfermedades concretas [105]. La HGMD es una colección exhaustiva de variantes de la línea germinal en genes nucleares que están asociadas con enfermedades hereditarias humanas y se recopila principalmente a partir de la literatura publicada. Sobre ClinVar se profundizará más adelante, pero a grandes rasgos permite archivar variantes brindadas directamente por los remitentes, junto a una interpretación de su patogenicidad a nivel variante-fenotipo [106]. Estas bases de datos son muy importantes en el proceso de interpretación, dado que aportan un punto de vista del usuario y generalmente, ClinVar, aporta información de la evidencia con la cual se generó la interpretación. Sin embargo, estas bases de datos a menudo contienen variantes clasificadas incorrectamente sin una revisión primaria de las pruebas, y a veces tienen registros contradictorios sobre la evaluación de la patogenicidad, y tienen medios limitados para resolver las diferencias entre usuarios [105]. Otro tipo muy relevante de bases de datos para la evaluación de patogenicidad son aquellas que aportan frecuencias alélicas de las variantes a gran escala en poblaciones ancestralmente diversas [105]. Un ejemplo de ello es la Genome Aggregation Database (gnomAD) [107], que contiene datos de genomas y exomas obtenidos de proyectos de secuenciación a gran escala. El conjunto de datos en su versión 2.1.1 para la referencia GRCh37/hg37 abarca 125.748 secuencias de exomas y 15.708 secuencias de genomas completos de individuos no emparentados, los cuales fueron secuenciados como parte de diversos estudios genéticos poblacionales y específicos de enfermedades. Esta cifra duplica con creces la de su precursor, el conjunto de datos del Exome Aggregation Consortium (ExAC) [76], que contiene los datos de exoma de 60706 individuos [76]. Al proporcionar estimaciones refinadas de la frecuencia alélica en poblaciones, estos recursos permiten a los investigadores utilizar umbrales de frecuencia alélica específicos de la ascendencia y la enfermedad para clasificar las variantes como fue explicado a través de las reglas [44].

2.1. Herramientas de clasificación usando *Machine Learning*

El uso de la inteligencia artificial (IA) ha hecho importantes avances en la asistencia sanitaria, y se está desarrollando una nueva clase de métodos de interpretación del genoma en general, pero también con la promesa de eliminar el cuello de botella de la interpretación para el diagnóstico de enfermedades genéticas raras [108, 109, 110, 111, 112, 113]. La priorización de variantes que explican

el fenotipo de una enfermedad resulta crucial para el diagnóstico genético de trastornos mendelianos raros. En este sentido se han desarrollado varias estrategias para priorizar las variantes patogénicas asociadas a trastornos raros las cuales se basan en estrategias de *Machine Learning* (ML) [105, 114]. Las recientes mejoras algorítmicas y de hardware, combinadas con datos a gran escala, han hecho posible que los métodos de ML alcancen resultados muy buenos en una amplia gama de tareas [105]. Muchos de estos métodos se han aplicado con éxito a una amplia variedad de datos genómicos, en particular debido al gran tamaño de los conjuntos de datos y a la complejidad de los mismos [115]. A continuación se plantearán algunos ejemplos de algoritmos basados en ML utilizados para la priorización de variantes patogénicas o la predicción de los efectos de variantes y su asociación con enfermedades

Algunos métodos como NCBoost [116] o ReMM [117] tienen un enfoque en la priorización de variantes patogénicas en regiones no codificantes del genoma según su impacto en la expresión génica y regiones reguladoras, dada la alta asociación de estas con enfermedades Mendelianas. En el caso de ReMM, se entrenó un clasificador *random forest* en un conjunto curado de 406 SNVs, incluyendo SNVs de ARN no codificantes largos. Se consideraron 26 características, incluyendo predictores de conservación y características epigenéticas. A pesar de la simplicidad del modelo, ReMM resultó valioso para priorizar variantes de enfermedades mendelianas cuando se integraron en un marco más amplio que consideraba regiones reguladoras candidatas y la relevancia fenotípica de los genes asociados [117]. Por otro lado, NCBoost amplía la propuesta de ReMM. Se trata de un un clasificador de SNVs no codificantes basado en árboles de decisión (GBDT[1]), en un conjunto curado de SNV no codificantes patogénicos asociados con genes de enfermedades mendelianas monogénicas y en SNV no codificantes comunes sin afirmaciones clínicas [116]. En los últimos años, las redes neuronales profundas han dado lugar a múltiples avances en diversas áreas de aplicación, y la genómica es una de ellas. A través de este enfoque se pueden identificar patrones complejos en los datos genómicos que pueden no ser detectables utilizando métodos tradicionales. Estudios realizados en 2015 demostraron la aplicabilidad de las redes neuronales profundas (DNN[2]) a los datos de secuencias de ADN [118]. Los modelos que pueden predecir fenotipos moleculares directamente a partir de secuencias biológicas pueden utilizarse como herramientas para evaluar las asociaciones entre variación genética y variación fenotípica. Éstos han surgido como nuevos métodos para la identificación de loci de rasgos cuantitativos (QTL[3]) y la priorización de variantes. En este sentido, hay varios modelos que usan estrategias de aprendizaje profundo (DL[4]) basados en secuencia que, en parte, pueden usarse como herramientas para evaluar el impacto que pueden tener determinadas variantes, siendo prometedor para encontrar posibles vías a fenotipos complejos. La utilización de la secuencia en estos métodos se realiza a través de distintas estrategias para la obtención de características, entre las cuales se encuentra el uso de herramientas de procesamiento de lenguaje natural (NLP[5]), por ejemplo, combinando *word embeddings* y *k-mers* a partir de la secuencia con otros métodos para extraer características para su clasificación. Uno de los algoritmos basados en secuencias de ADN es "DeepSEA", que utiliza una DNN para predecir los efectos funcionales de las variantes genéticas no codificantes de novo [119, 120]. El algoritmo aprende directamente un código de secuencia reguladora a partir de datos de perfiles de cromatina a gran escala, lo que permite predecir los efectos relacionados con la cromatina de las alteraciones de secuencia con una sensibilidad de un solo nucleótido. Además, esto ha permitido la priorización de las variantes funcionales, incluidos los loci de rasgos cuantitativos de expresión (eQTL) y las variantes asociadas a enfermedades. A partir de allí se ha disparado el número de publicaciones que describen la aplicación de las redes neuronales profundas a la genómica mientras que en paralelo la comunidad del aprendizaje profundo ha mejorado sustancialmente la calidad de los

[1] *Gradient Boosting Decision Tree*
[2] Deep Neural Network
[3] *Quantitative Trait Loci*, en inglés.
[4] Deep Learning
[5] *Natural Language Processing*, en inglés.

métodos y ha ampliado su repertorio de técnicas de modelado [118]. Un ejemplo de trabajo actual en este campo, y que tiene como antecedentes a herramientas como "Basenji" [121] o "ExPecto" [122] (ambos basados en redes neuronales convolucionales [CNN]), es "Enformer", un método basado en Transformers (una clase de modelo de DL) para predecir la expresión génica y estado de la cromatina. Este método logra integrar información de interacciones de largo alcance (hasta 100 kb de distancia) en el genoma [123].

En el caso de las variantes codificantes, diversas herramientas pueden predecir si la variante es perjudicial para la función o la estructura de la proteína utilizando distinto tipo de información: información evolutiva, el contexto dentro de la secuencia proteica y propiedades bioquímicas. Estos métodos *in silico* incluyen sistemas de puntuación individual, y usan estrategias para priorizar las variantes patogénicas asociadas a trastornos. Muchos de ellos están basados en la utilización de algoritmos de *machine learning*, tales como CADD [124], DANN [125], FATHMM-MKL [126], M-CAP [127] y REVEL [128], así como meta-predictores, como Condel [129] y MetaSVM [130]. Muchos tienen una base teórica similar, pero también tienen limitaciones conocidas, como una precisión moderada, baja especificidad y sobrepredicción. En el caso de CADD (*Combined Annotation-Dependent Depletion*) [124] [131], es un metodo que integra originalmente aproximadamente 63 características que provienen de la anotación del genoma y brinda una puntuación a cualquier posible SNV o INDEL, utilizando **support vector machines** (SVM) con *kernel* lineal. En versiones posteriores de la herramienta, algunas adaptaciones implicaron un cambio de modelo, pasando a entrenar un modelo de regresión logística regularizada [132]. CADD hoy en día es una de las herramientas de apoyo a la clasificación con un peso importante (asistida por otros aspectos). Otras herramientas como DANN [125], por ejemplo, usan DL para capturar relaciones complejas entre características de entrada y patogenicidad, permitiendo priorizar variantes tanto codificantes como no codificantes. DANN surge como una posible mejora a las limitaciones de la primera versión de CADD, considerando las limitaciones impuestas por el modelo lineal utilizado [125]. Otras herramientas como FATHMM (*Functional Analysis Through Hidden Markov Models*), fueron modificando su enfoque. En este caso originalmente la herramienta usaba una herramienta de modelos de markov ocultos (HMM[6])[133], y en una versión más actual, llamada FATHMM-MKL [126] que predice consecuencias funcionales de las variantes de secuencias codificantes y no codificantes, mediante el uso de *multiple kernel learning* (MKL). M-CAP, por otro lado, se suma a los algortimos que identificaron la potencialidad de GBDT para la clasificación de patogenicidad, en este caso aplicado a variantes raras *missense* en el genoma humano. M-CAP se basa en la combinación de *scores* de patogenicidad anteriores (incluyendo SIFT, Polyphen-2 y CADD) con nuevas características, y se demostró una reducción en una lista típica de variantes VUS de exoma/genoma de 300 a 120 [127].

Los algoritmos mencionados anteriormente, tienen en común la utilización de información sobre el genotipo (secuencia y atributos genómicos) para proporcionar una predicción de la patogenicidad de las variantes. Sin embargo, dado que cada persona sana suele albergar unas 100 variantes deletéreas con pérdida de función, varios trabajos consideraron necesario tener más en cuenta la asociación genotipo-fenotipo para las aplicaciones clínicas. Para priorizar aún más las variantes, se han propuesto métodos basados en el fenotipo, tales como Phen-Gen [134], eXtasy (basado en algoritmo *random forest*) [135] o Exomiser [136]. Estos métodos combinan los resultados de los algoritmos de predicción *in silico* existentes y una medida de relación fenotípica, para la puntuación y clasificación de las variantes genéticas causantes de enfermedades. En el mismo contexto, en 2019 se publicó "Xrare", un modelo de priorización que se basa en una amplia base de datos de variantes como conjunto de entrenamiento, que tiene su enfoque principal en la unificación de características genómicas con fenotípicas [4]. Con este método se busca aportar a mejorar la tasa insatisfactoria de diagnóstico en entornos clínicos reales, a través del enfoque en solucionar el carácter incompleto, la heterogeneidad,

[6] *Hidden Markov Model*

la imprecisión y el ruido de las descripciones del fenotipo de la enfermedad descritos hasta el momento. Para cumplir con el objetivo se desarrolla en conjunto con Xrare, un mecanismo de puntuación de similitud fenotípica, y ambos permiten modelar en conjunto las características fenotípicas y características genéticas, incluidas las características basadas en las reglas ACMG/AMP, necesarias para una priorización precisa. El trabajo desarrolla un *score* de fenotipo denominado Contenido de Información de Emisión-Recepción (ERIC), el cual permite medir similitud fenotípica entre los fenotipos imprecisos y ruidosos de los pacientes y los fenotipos conocidos asociados a una enfermedad o un gen, y luego usar este *score* como característica en los algoritmos. Se demostró que ERIC se situó por encima de otros *scores* de similitud fenotípica en presencia de fenotipos imprecisos y ruidosos, y varias simulaciones y datos clínicos reales demostraron que Xrare supera a los métodos alternativos existentes en un 10-40 % en varios escenarios de diagnóstico mediante variantes [4]. El modelo consiste en GBDT, y se entrena usando principalmente variantes de ClinVar, lo que resulta una base e inspiración principal para el trabajo actual, dado el enfoque que presenta. Otro método de apoyo al diagnóstico de enfermedades raras es "Fabric GEM" [112], una herramienta de apoyo a la toma de decisiones clínicas basada en IA para agilizar la interpretación del genoma, mediante la priorización de genes asociados a enfermedades. GEM recibe variantes en formato VCF y metadatos de casos, incluidos los fenotipos del paciente (probando) en forma de términos de la Ontología de Fenotipos Humanos (HPO). Fue entrenado en una cohorte retrospectiva 119 probandos, en su mayoría neonatos en cuidados intensivos, diagnosticados con ER, que recibieron WES y/o WGS. El método clasificó más del 90 % de los genes causales entre los principales o segundos candidatos y priorizó para su revisión una media de 3 genes candidatos por caso, utilizando descripciones de fenotipos curadas [112].

A pesar de los éxitos del uso de ML, las aplicaciones de estas estrategias en general en medicina se enfrentan a varios retos particulares. Uno de los más importantes es la escasez de ejemplos etiquetados fiables Las etiquetas de los datos suelen proceder de profesionales que pueden no estar seguros de sus clasificaciones o discrepar con otros expertos. Además, dado que la recolección de estos conjuntos de datos puede requerir tiempo por parte de los evaluadores, el etiquetado de cantidades de datos puede resultar prohibitivamente caro [105].

Algunos de los retos pueden abordarse con otros enfoques informáticos. Por ejemplo, los avances en el procesamiento del lenguaje natural han permitido el desarrolla de estrategias novedosas en el área. La literatura biomédica es un recurso rico para obtener información clave sobre variantes genómicas porque la mayoría de los nuevos hallazgos en investigación biomédica se publican y comparten a través de revistas revisadas por pares o publicaciones en congresos e indexadas en bases de datos bibliográficas, como PubMed [137] o PubMed Central (PMC) [138] [139]. El diagnóstico de los trastornos mendelianos requiere una laboriosa investigación bibliográfica. Los clínicos formados en pueden pasar horas buscando la publicación o publicaciones adecuadas que apoyen el gen que mejor explica la enfermedad de un paciente, dado que proceso de curación manual requiere no solo de evaluadores altamente cualificados, sino que es caro y requiere mucho tiempo [139]. En este sentido también se han realizado una serie de aportes basados en la literatura, que basados en NLP permiten acelerar estos procesos y sirven como soporte al evaluador. Un ejemplo es "AMELIE" (*Automatic Mendelian Literature Evaluation*), que a través del análisis de ≈29 millones de resúmenes de PubMed y la descarga y análisis de cientos de miles de artículos de texto completo busca información que respalde la causalidad y los fenotipos asociados de la mayoría de las variantes genéticas publicadas [110] [102]. La herramienta logra priorizar las variantes candidatas de los pacientes en función de su probabilidad de explicar un determinado conjunto de fenotipos. AMELIE clasificó el gen causante en primer lugar para el 66 % de los 215 pacientes únicos (sin exomas de familiares) mendelianos diagnosticados del proyecto Deciphering Developmental Disorders [140]. Además, la evaluación de sólo los 11 genes con mayor puntuación en AMELIE de 127 genes candidatos por paciente en total dio como resultado un diagnóstico rápido en más del 90 % de los casos. La evaluación de todos los casos basada en AMELIE fue entre 3 y 19 veces más eficiente que los enfoques basados en bases de datos

seleccionadas manualmente [110]. Otro ejemplo es "AVADA" (Automatic VAriant evidence DAtabase)(AVADA), que utiliza NLP para identificar automáticamente evidencias de variantes genéticas patogénicas en literatura primaria de texto completo sobre enfermedades monogénicas y convertirlas en coordenadas genómicas. Utiliza 47 expresiones regulares para reconocer variantes genómicas,. También emplea una herramienta personalizada de reconocimiento de nombres de genes que aprende y utiliza listas de nombres de genes obtenidas del Comité de Nomenclatura Genética HUGO (HGNC) [141] y de la base de datos UniProt [142] para encontrar genes candidatos asociados a variantes en un artículo. Se demostró que AVADA logra recuperar automáticamente casi el 60 % de las posibles variantes causantes de enfermedades depositadas en la *Human Gene Mutation Database* (HGMD) [143], lo que supone una mejora de 4.4 veces respecto al mejor extractor automatizado de variantes de código abierto[111]. Otro tipo de aplicación de NLP en el diagnóstico es el desarrollado en 2019 por *M. Clark et al.* [144], en donde se trabajó en una plataforma para el diagnóstico de enfermedades genéticas con fenotipado e interpretación automatizados, en un contexto de unidades neonatales y centros de cuidados intensivos pediátricos. Para ello se usó información genómica y procesamiento de lenguaje natural clínico (CNLP) para extraer los fenotipos de los niños a partir de registros médicos electrónicos con un 80 % de precisión y un 93 % de recuperación. En 101 niños con 105 enfermedades de base genética, una media de 4.3 características fenotípicas extraídas de CNLP coincidieron con las características fenotípicas esperadas de esas enfermedades, en comparación con una coincidencia de 0.9 características fenotípicas utilizadas en la interpretación manual [144].

Capítulo 3

Materiales y métodos comunes

Las dos grandes partes del presente trabajo se valen de datos de variantes gnómicas. Dado este aspecto en común es que estos datos se presentan en esta instancia. Los datos de variantes se obtienen principalmente de bases de datos publicas, y en instancias de prueba y desarrollo se contó con la posibilidad de incluir datos de muestras provenientes de URUGENOMES. Los datos de base se obtienen en formato VCF, es decir que no fue necesario realizar el procesamiento de la secuencia que implique un llamado de variantes para esta instancia. En la Figura B.1 se presenta una descripción gráfica de cómo se ve un archivo VCF a grandes rasgos [71]. Luego estos archivos son procesados a los efectos de adquirir las características necesarias de las variantes para su procesamiento. A continuación se dará una descripción de la principal base de datos usada en este trabajo que provee las variantes, además del proceso de anotación en común que sufren los datos. Cabe destacar que en este capítulo no se encuentra la metodología completa llevada a cabo, sino únicamente la que se encuentra en común entre la plataforma y el proceso de clasificación. El procesamiento específico realizado para cada parte se presentará en los capítulos correspondientes.

3.1. ClinVar

ClinVar es el principal repositorio público de relaciones entre las variaciones en el genoma humano y los fenotipos asociados a las mismas, con aporte de evidencia a dichas relaciones. Los laboratorios clínicos, investigadores y expertos comparten en este repositorio sus propias clasificaciones de variantes identificadas en pacientes, aportando evidencia de dicho hallazgo, y documentando el impacto clínico de la mutación. Este recurso fue desarrollado con el objetivo de satisfacer la necesidad de acceder de una forma interactiva y sistemática a la interpretación clínica de la variación genómica humana, y es mantenida por el NCBI y se encuentra relacionada a múltiples recursos del NCBI, tales como *dbSNP*, *PubMed Central* o *Reference Sequence Database* [145, 146, 147, 148]. ClinVar es una base de datos no curada que almacena la información proporcionada por sus usuarios sobre variantes encontradas en pacientes y las afirmaciones realizadas con respecto al impacto clínico de estas variantes, además de datos extra que puedan aportar respaldo al reporte realizado. El repositorio de ClinVar fue iniciado como un prototipo puesto en marcha de forma oficialmente en el año 2013, y desde entonces aporta actualizaciones periódicas y constantes, brindando a sus usuarios el acceso a un amplio conjunto de interpretaciones cínicas actuales e históricas [106].

El contenido de ClinVar se puede dividir en 5 grandes categorías:

- Usuario: ClinVar acepta información de variantes aportadas tanto por organizaciones o en condición de usuarios individuales, con la misma identificada a través de pruebas clínicas, investigación y curación de literatura.
- Variante: un dato clave en el modelo de datos de ClinVar es la variación y la representación de su relación con un fenotipo determinado. La variación se almacena como la secuencia en una ubicación determinada o como una combinación de cambios en la secuencia en múltiples ubicaciones. ClinVar utiliza los identificadores aportados por otras bases de datos, en el caso

de que la localización de la variación sea conocida, o ingresa la variante a las mismas para que les sea asignado un identificador en el caso de que las variantes identificadas sean nuevas.

- Fenotipo: ClinVar representa a los fenotipos como un único concepto o como un conjunto de conceptos.
- Interpretación: se representan interpretaciones de la importancia clínica, brindada por el usuario, utilizando los términos de relevancia clínica recomendados por el ACMG [44].
- Evidencia: datos que apoyan la interpretación de la relación variante-fenotipo. Generalmente la evidencia consiste en la descripción de cómo fueron obtenidas las variantes y en qué contexto, pudiendo ser representada como un número de observaciones por persona o cromosoma, número de segregaciones observadas, número de veces que se identificaron otras variaciones raras en el mismo gen u otros genes, etc.

En el caso del presente trabajo, las interpretaciones de las variantes (con peso de evidencia significativo) serán la etiquetas que guiarán los algoritmos supervisados. Esto está representado en el campo "significado clínico" de la base de datos.

ClinVar usa para el significado clínico términos estándar como también términos no-estándar. Los términos estándar consisten en aquellos sugeridos para enfermedades Mendelianas por la ACMG [44]. En la Tabla 3.1 se listan las opciones válidas más relevantes o frecuentes para el significado clínico y su definición.

Valor de significado clínico	**Descripción para el uso sobre registros**
Benigno	Según lo recomendado por ACMG/AMP para variantes interpretadas como desórdenes mendelianos.
Probablemente benigno	Según lo recomendado por ACMG/AMP para variantes interpretadas como desórdenes mendeliano.
Significado incierto	Según lo recomendado por ACMG/AMP para variantes interpretadas como desórdenes mendelianos.
Patogénico	Según lo recomendado por ACMG/AMP para variantes interpretadas como desórdenes mendelianos. Las variantes con penetrancia baja se consideran como patogénicas
Respuesta a drogas	Variantes que afectan la respuesta a fármacos, no causan enfermedades.
Asociación	Variantes identificadas en un estudio de GWAS e interpretadas con significancia clínica.

Factor de riesgo	Variantes que si bien no causan un desorden, aumentan el riesgo
Protector	Variantes que disminuyen el riesgo a un desorden.
Afecta	Variantes que causan un fenotipo no patológico.
Datos conflictivos de los remitentes	Variantes reportadas por asociación, en la que los grupos tienen interpretaciones conflictivas sobre la variante pero aportan sólo un significado.
Otro	Variantes para las cuales ClinVar no tiene un término apropiado para la entrada.
No provisto	Variantes sin interpretación de significado clínico. Se limita a: entradas que reportan una publicación sobre una variante, sin interpretación del significado clínico; entradas que tienen un significado funcional pero no significado clínico; entradas provistas por médicos con información sobre los individuos con la variante, pero sin interpretación del significado clínico.

CUADRO 3.1: Nomenclatura para el significado clínico y su correspondiente definición en la base de datos ClinVar [106]. Los significados clínicos corresponden a una combinación entre aquellos propuestos por ACMG/AMP, y otros aportados, por ejemplo, por ClinGen mediante sus propios lineamientos.

Como fue mencionado anteriormente, ClinVar contiene variantes cuyo significado clínico no se encuentra definido, y que debido a la falta de evidencia no se llegue a un consenso sobre su significado. Dado que es una base de datos no curada que no cuenta con mecanismos para resolver estos conflictos, si hay diferencias en la interpretación entre los remitentes de los reportes de ClinVar entre los 5 niveles de significado clínico recomendados por ACMG [44], el significado se reporta como un conflicto, usando la etiqueta *"Conflicting_intepretations_of_pathogenicity"*. En la Tabla 3.2 se enumeran casos de combinación de valores de interés.

Combinación de valores de significado clínico	**Reporte**
Patogénico y probablemente patogénico	"Patogénico/Probablemente patogénico"

(Patogénico o probablemente patogénico o Benigno o probablemente benigno) y Significado incierto	"Interpretación conflictiva de patogenicidad"
(Patogénico o probablemente patogénico) y (benigno o probablemente benigno)	"Interpretación conflictiva de patogenicidad"
Benigno y probablemente benigno	"Benigno/Benigno probablemente benigno"
Cualquier valor de ACMG y cualquier valor no-ACMG Ej.: significado incierto y factor de riesgo	"Valor ACMG/valor no-ACMG" ó "Significado incierto, factor de riesgo"
Valores conflictivos de ACMG y valor no-ACMG Ej.: Patogénico y Significado incierto y factor de riesgo	"Interpretación conflictiva de patogenicidad, valor no-ACMG" ó " Interpretación conflictiva de patogenicidad, factor de riesgo"
Valor no-ACMG y múltiples valores no-ACMG	"valor no-ACMG,valor no-ACMG"
Un grupo no está de acuerdo con el valor ACMG	"datos conflictivos de los remitentes"

CUADRO 3.2: Tabla de combinación de valores de significado clínico y su reporte correspondiente. Cuando los usuarios en ClinVar difieren en las clasificaciones emitidas, ClinVar no cuenta con mecanismos para generar etiquetas consenso o eliminar conflictos generados, por lo que los valores son combinados. Dependiendo del tipo de etiqueta (ACMG o no ACMG) y/o del nivel de discrepancia las etiquetas pueden ser agregadas o se generan etiquetas conflictivas.

Otro campo de ClinVar que es de relevancia, es el estado de revisión, que indica una idea de qué tan fundada está la clasificación emitida. Cada estado de revisión en la interfaz de ClinVar está directamente asociado a una cantidad de "estrellas", que brinda una idea visual. En la Tabla 3.3 muestra la relación entre cada categoría de estado de revisión, su descripción y el número de estrellas que le corresponde [149].

Código de estrellas	**Estado de Revisión**	**Descripción**
★★★★★	"*practice guideline*"	Guía práctica aportada por el solicitante o quien reporta la variante.
★★★★☆	"*reviewed by expert panel*"	Revisado por un panel de expertos.
★★☆☆☆	"*criteria provided, multiple submitters, no conflicts*"	Dos o más solicitantes con criterios de acierto y evidencia para la misma interpretación.

★☆☆☆☆	"*criteria provided, conflicting interpretations*"	Varios solicitantes aportan criterios de acierto y evidencia pero con distintas interpretaciones.
★☆☆☆☆	"*criteria provided, single submitter*"	Un sólo solicitante aporta interpretación con evidencia y criterio de acierto.
☆☆☆☆☆	"*no assertion for the individual variant*"	El alelo no tiene interpretación por ningún solicitante. La entrada en ClinVar es como un componente de un haplotipo o fenotipo.
☆☆☆☆☆	"*no assertion criteria provided*"	El alelo fue incluido en una solicitud con interpretación pero sin criterio de acierto ni evidencia.
☆☆☆☆☆	"*no assertion provided*"	El alelo fue incluido en una solicitud que no provee interpretación alguna.

Cuadro 3.3: Estado de revisión aportado por ClinVar con su correspondencia en número de estrellas en la página web [149] [150].

Los datos de ClinVar fueron descargados en distintos formatos de texto plano (VCF y TXT), a partir del sitio FTP disponible en el sitio oficial de la base de datos[151]. Los mismos corresponden al genoma de referencia GRCh37.

3.2. Anotación de características biológica

La anotación de datos en esta instancia se hace con ANNOVAR. ANNOVAR es una herramienta que permite anotar las variantes con múltiples bases de datos, asignándoles, en parte, su consecuencia funcional con respecto a genes [93]. También permite, por ejemplo, inferir bandas citogenéticas, reportar *scores* de efecto funcional, encontrar variantes en regiones conservadas, entre otras funciones. Además de las anotaciones de efectos funcionales, ANNOVAR presenta distintas funciones para distintos casos de análisis deseados. Por ejemplo, presenta la capacidad de realizar anotaciones basadas en regiones genómicas así como también realizar comparaciones de variantes entre bases de datos existentes. ANNOVAR permite 3 tipos o modos de anotación de las variantes:

- *Gene-based annotation:* identifica si las variantes causan cambios en la codificación de proteínas y los aminoácidos que se ven afectados.
- *Region-based annotation:* identifica variantes en regiones genómicas específicas, por ejemplo, regiones conservadas entre 44 especies, sitios de unión de factores de transcripción predichos, regiones de duplicación segmentaria, observaciones de GWAS, base de datos de variantes genómicas, etc.
- *Filter-based annotation:* identificar las variantes que están documentadas en bases de datos específicas, por ejemplo, si una variante está reportada en dbSNP, cuál es la frecuencia alélica en el Proyecto 1000 Genomas, ESP 6500 exomas o Exome Aggregation Consortium (ExAC) o Genome Aggregation Database (gnomAD), calcular los *scores* de predición *in silico* SIFT, PolyPhen,

LRT, MutationTaster, MutationAssessor, FATHMM, MetaSVM, MetaLR, entre muchos otros recursos más.

En la figura 3.1 se muestra el flujo de procesamiento de ANNOVAR, y cómo se utiliza cada método incluido en su funcionamiento. ANNOVAR es una herramienta escrita en lenguaje Perl [152] y se utiliza a través de la línea de comandos, en cualquier sistema operativo (que tenga un intérprete de Perl instalado). En el caso de este trabajo se utilizó la versión libre de la herramienta dado su uso no-comercial esperado. Como se puede observar, el primer paso antes de aplicar la anotación se procede a la descarga de las bases de datos requeridas. En la siguiente sección se presentará un detalle de las bases de datos que fueron interrogadas para realizar la anotación. Las bases de datos deben adquirirse explicitando el ensamblaje de referencia, y todas las bases de datos deben hacer referencia al mismo ensamblaje con el que fueron adquiridas las variantes, en este caso GRCh37/hg37/hg19 . Una vez que se cuenta con las bases de datos actualizadas, se procede a aplicar la anotación. En el caso del trabajo actual, se realizó la anotación a través del flujo de trabajo básico presentado en la Figura 3.1, en donde el conjuntos de variantes fue ingresado en formato VCF y se aplicó la anotación usando `table_annovar.pl`. Dado que el funcionamiento de ANNOVAR depende exclusivamente del tipo de datos específico de la herramienta (`avinput`), dicho método se encarga de hacer la conversión de archivos. Solamente en un caso se requiere la conversión explícita de los datos, que es en el cálculo de uno de los scores usados posteriormente (*Grantham Score*). Ambos procesamientos se hacen en simultáneo en este caso. `table_annovar.pl` es un *wrapper* en torno a `annotate_variation.pl`, que es el núcleo de funcionamiento de ANNOVAR, el cual implementa los tres tipos de anotación mencionados anteriormente, sobre los conjuntos de datos descargados inicialmente. Dada la modalidad en la que se instanció, la salida se puede obtener en distintos formatos de texto: VCF, TSV o CSV. La salida contiene los datos incorporados, en el caso de tratarse de los formatos TSV o CSV, en forma de nuevas columnas, y en el caso de tratarse de un VCF, como nuevos valores en el campo "INFO" del mismo [71]. En este trabajo en particular se procedió a obtener una tabla final separada por tabulaciones, a los efectos de su adaptación con el resto del flujo de trabajo y su incorporación a la base de datos del proyecto (a tratar en el próximo capítulo).

3.2.1. Bases de Datos interrogadas

En el presente trabajo se realizó la anotación en los tres modos mencionados anteriormente (*gene-based, filter-based* y *region-based*). A continuación se listarán las bases de datos interrogadas, agrupadas según el tipo de anotación:

- *Gene-based annotation*:
  - `RefSeq Gene` [148] (v2021.10.19): se obtiene como una porción de la base de datos RefSeq, brindando información de la región en la que se se encuentra la variante, el gen en el que se encuentra y detalles del mismo, el cambio de aminoácido generado, y el efecto funcional de la variante.
- *Region-based annotation*:
  - `cytoBand`: agrega información sobre la banda citogenética en la que se encuentra la variante.
  - `genomicSuperDups`: agrega información que permite identificar regiones segmentarias.
  - `gwasCatalog`: agrega información de la variante en caso de que haya sido reportada en estudios de asociación (GWAS)previamente.
- *Filter-based annotation*:

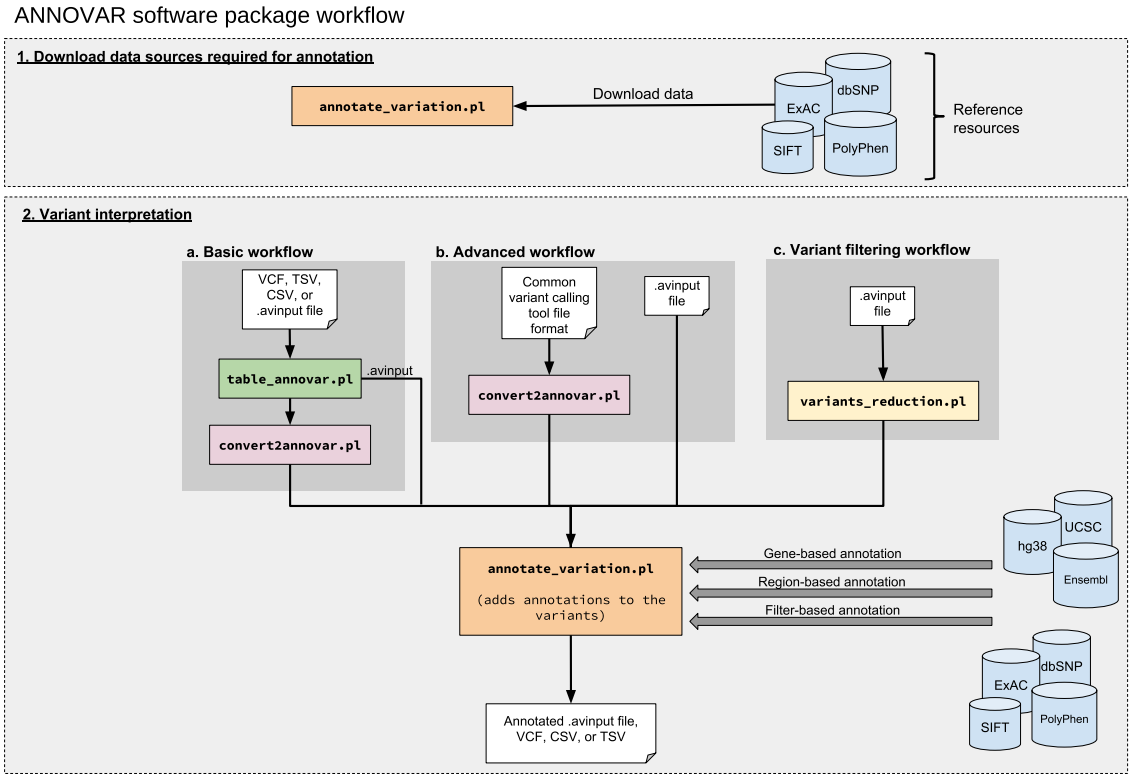


FIGURA 3.1: Flujo de funcionamiento de ANNOVAR. En una primera instancia de descargan las bases de datos de diversos recursos. Luego la anotación se puede hacer en tres formas distintas, que ofrecen diferentes funcionalidades pero también flexibilidad en su uso: un flujo de trabajo básico, avanzado y de filtro. Extraído de [153].

- `ClinVar` [106] (v2021.05.01): agrega información proveniente de los aportes realizados en ClinVar respecto a la variante respecto al significado clínico final reportado, estado de revisión final, identificador de la variante en ClinVar, identificadores en otras bases de datos. En el caso del proyecto actual las variantes en común ya provienen de ClinVar, por lo que dicha información de obtiene directamente de dicha base de datos, sin embargo el proceso de anotación fue estandarizado para procesar variantes que no provengan de ClinVar también.
- `ESP 6500` [74] (v2014.12.22): brinda la frecuencia del alelo alternativo en todas las muestras presentes en el proyecto *Exome Sequencing Project* (ESP), incluyendo SNPs e INDELs.
- `1000 genomas` [73] (v2015.08.24): brinda la frecuencia del alelo alternativo en todas las variantes presentes las 2504 muestras del proyecto en las poblaciones: europea, asia del este, africana, americana, y total.
- `dbSNP138` [146]: asigna a cada variante un RSID proveniente de la base de datos dbSNP138 además .
- `ExAC` [76] (v2015.11.29): contiene la frecuencia alela proveniente de variantes de 65000 exomas completas, para las poblaciones: africana, americana, asiática del este, finlandesa, europeos no-finlandeses, asia del sur, además de otros y el total.
- `Reg-SNP intron` [154] (v2018.09.20): prioriza variantes intrónicas que son potencialmente deletéreas.

- `dbNSFP` [155](v2018.09.21): incluye la predicción de varios *scores* de patogenicidad ne variantes codificantes, incluidos SIFT, PolyPhen2 HDIV, PolyPhen2 HVAR, LRT, MutationTaster, MutationAssessor, FATHMM, PROVEAN, MetaSVM, MetaLR, VEST, MCAP, CADD, GERP++, DANN, fathmm-MKL, Eigen, GenoCanyon, fitCons, PhyloP and SiPhy [155].
- `GWAVA` [156] (v2015.06.23): brinda el *score* GWAVA que prioriza variantes patogénicas en regiones no codificantes del genoma.
- `gnomAD genoma y exoma` [107] (v2.1.1): aporta tanto para genoma como exoma las frecuencias de los alelos alternativos en las poblaciones africana, sur-asiática, americana, asiática del este, europea no-finlandesa, finlandesa, judia ashkenazi, además de brindar frecuencias en población femenina y masculina, total, y otros.
- `Kaviar` [157] (v2015.12.03): frecuencia alélica asociada a 170 millones de variantes provenientes de 13000 genomas y 64000 exomas a partir de 43 proyectos [157].
- `Eigen` (v2016.03.30) [158]: agrega a cada variante el *score* de predicción de impacto funcional Eigen.
- `dbscsnv11` (v1.1): agrega a cada variante la predicción del efecto de la variante en el sitio de *splicing*.
- `Intervar` [86] (v2018.03.25): Agrega la interpretación de cada variante según las 28 reglas ACMG/AMP (soporta únicamente SNVs).

Capítulo 4

Plataforma de aprendizaje

Parte del presente trabajo consistió en la implementación de una plataforma web destinada, principalmente, a la recolección de datos asociados a la clasificación de variantes cortas en el genoma humano. Este espacio permitiría que a largo plazo, los algoritmos de clasificación puedan tener como insumo datos y evaluaciones que provengan de expertos de la región en el área. No obstante, para realizar estas evaluaciones, es importante contar con recursos que comprendan el flujo de trabajo implicado en el análisis de variantes genómicas y su contexto para valorar su posible impacto. Es por ello, que además, se trabajó en la elaboración de una plataforma que brinde un primer acercamiento al aprendizaje en la interpretación de dichas variantes, donde los usuarios puedan adquirir habilidades en la tarea a través de un flujo de trabajo controlado. Las funcionalidades expuestas implican que la plataforma permita centralizar información de múltiples recursos y bases de datos, y resumirla de forma adecuada y ordenada. Todo ello a los efectos de que el usuario comprenda las características más relevantes de la variante, para luego generar un veredicto sobre su patogenicidad. Esta información involucra datos sobre el efecto de codificación de la variante, su localización en el genoma, los genes a los que puede afectar, su frecuencia en la población, la función de la proteína asociada en el caso de localizarse en una región codificante, los fenotipos relevantes asociados, la literatura relacionada, los estudios clínicos y la patogenicidad determinados previamente, entre una amplia gama de datos más. Teniendo estos aspectos en cuenta, la plataforma generada pretende ser explorada como una herramienta con múltiples utilidades, las cuales tengan en común la evaluación de variantes a través de procesos estándar de clasificación y aplicación de reglas y directrices ya establecidas (como las ACMG/AMP). El público objetivo del sistema final tiene en común el desempeño en tareas de diagnóstico mediante datos de secuenciación masiva de genoma o exoma, o el aprendizaje en el mismo, el cual puede darse a distintos grados: en el contexto de la formación de recursos humanos biomédicos de diferentes orientaciones a través de la integración de la plataforma en espacios de aprendizaje, desde un usuario que tenga como objetivo aprender o interiorizarse en el proceso de forma individual, hasta un profesional que se desempeñe en la tarea a diario.

4.1. Especificaciones del Sistema Web

4.1.1. Alcance

La plataforma web a implementar debería contar con las funcionalidades básicas que permitan el etiquetado de variantes, su correcto almacenamiento, y un primer acercamiento al aprendizaje de usuarios. El desarrollo de la plataforma implica que desde el *backend* la implementación debería ser suficiente para obtener un producto mínimo viable (MVP[1]). Aún así se espera que los cimientos sean sólidos a los efectos de que en el futuro se pueda mantener la estructura general, y trabajar en las mejoras pertinentes de la interfaz de usuario. Esta última, a su vez se espera que sea suficiente como para ilustrar el concepto y los fundamentos, llegando hasta una instancia de prueba de concepto (PoC[2]).

[1] *Minimum Viable Product, en inglés.*
[2] *Proof of Concept, en inglés.*

4.1.2. Requerimientos funcionales

- Operatividad: el sistema debe permitir etiquetado de variantes y el entrenamiento básico de usuarios. Para este fin los usuarios deben poder acceder a la información de variantes de forma resumida y ordenada a través de una interfaz de requerimientos mínimos, detrás de la cual debe haber un sistema robusto de almacenamiento e intercambio de datos.

- Autenticación: tanto en el etiquetado de variantes como en el entrenamiento de usuarios implican que haya un registro de los mismos. La autenticación garantiza que los usuarios validen su identidad antes de realizar las funciones de la plataforma, permitiendo su seguimiento y categorización en el sistema de *scores* y rendimiento, además de actividades distintivas según el tipo de usuario. A los efectos de sentar la base para una plataforma futura con funcionalidades de usuario más detalladas, la autenticación es fundamental.

- Niveles de autorización: no todos los usuarios de la plataforma tendrán acceso a los mismos tipos de datos, solamente las personas adecuadas tendrán acceso a la manipulación de datos confidenciales, como por ejemplo, datos de pacientes de interés.

4.1.3. Requerimientos no funcionales

Excluyendo la elección del lenguaje de programación y el *framework* de desarrollo de la plataforma, los cuales serán discutidos más adelante, a continuación se enumeran algunos requerimientos no funcionales a tener en consideración:

- Asequibilidad: la plataforma debe ser web, gratuita y de código abierto. Además, debe ser accesible a través de todos los navegadores de internet estándar, a través de una URL estable. Debe contemplarse la posibilidad de que se acceda a la aplicación desde distintos sistemas operativos.

- Adaptabilidad: como fue mencionado previamente, el sistema puede ser utilizado para un mínimo de dos casos de uso, por lo que debe ser adaptable a cada uno de ellos, y a posibles casos de expansión.

- Integrabilidad: el sistema debería estar preparado para permitir el uso por parte de aplicaciones de terceros, ofreciendo *endpoints* desde una API [3].

- Internacionalización: si bien la plataforma eventualmente puede estar disponible en varios idiomas, con la posibilidad de añadir idiomas de forma sencilla, la misma debe encontrarse originalmente en español, a los efectos de mostrar cierta identidad nacional y representación de países de habla hispana. En esta instancia no se desarrollará soporte para múltiples idiomas.

- Seguridad: las funcionalidades principales están disponibles sólo para los usuarios autenticados. Habrá funcionalidades divididas por roles de usuario que cuenten con diferentes niveles de autorización. Solamente los administradores son capaces de crear, copiar, ver, cambiar o borrar información de la base de datos.

- Usabilidad: la aplicación debe ser fácil de usar por personas que cuentan con manejo de herramientas informáticas en diferentes grados, por lo que el sistema no debe requerir instalación de herramientas por fuera de las de uso regular, a los efectos de no añadir dificultades o restricciones en el uso. En el sentido de la aplicación particular, la plataforma debe ser abordable y entendible tanto por expertos en el ámbito de la clasificación de variantes genómicas como por usuarios que se encuentren iniciando en la disciplina.

[3] Application Programming Interface, en inglés

4.1.4. Actores

Los actores que participan en el sistema, son:

- Usuario común: consiste en los individuos que, por un lado, se beneficiarán de los servicios que aporta la plataforma, y en los que aporten un servicio a la misma desde diferentes aspectos. Si bien los roles de los usuarios cambian a medida que avanzan de nivel, se distinguen tres grandes tipos de usuarios según su nivel de participación y especialización:
  - Usuario estándar: realiza las funcionalidades estándar de la plataforma de aprendizaje y clasificación con variantes provenientes de bases de datos públicas.
  - Usuario estándar con permisos: además de la posibilidad de realizar las funcionalidades estándar de la plataforma tiene habilitadas variantes provenientes de pacientes de interés particular.
  - Usuario invitado: solamente requiere de la plataforma como servicio y no participa en el aprendizaje ni en la clasificación.
- Usuario administrador: participa en la actualización del sistema (bases de datos, anotación, etc.) y en la corrección de errores inmediatos.

4.1.5. Casos de uso

La primera versión de la plataforma debe contemplar al menos cuatro casos de uso: registro y login de usuario, entrenamiento de usuarios, etiquetado de variantes y clasificación de nuevas variantes. Dada la dinámica del trabajo, en esta versión se consideró agregar un caso de uso más: la clasificación de variantes de pacientes. Los casos de uso se ilustran en la Figura 4.1, utilizando un diagrama UML[4] para resumir la interacción de los actores con el sistema.

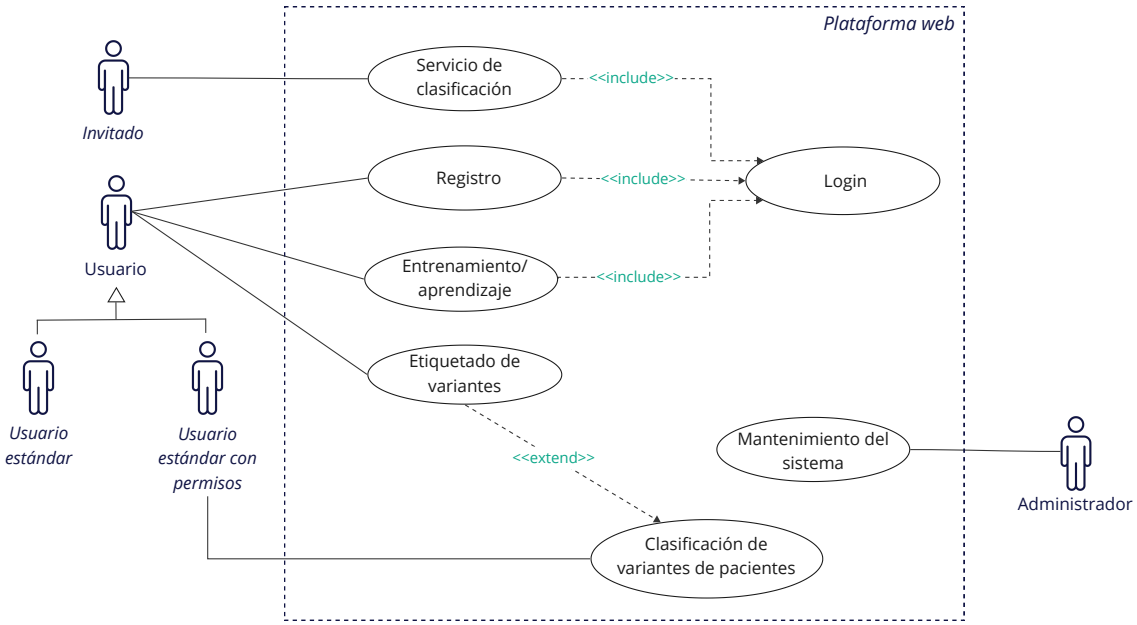


FIGURA 4.1: Diagrama UML de comportamiento según los casos de uso en la plataforma web. Los cuatro tipos de usuarios (invitado, estándar, estándar con permisos y administrador) interactúan con la plataforma web a través de cuatro casos de uso: registro y login de usuario, entrenamiento de usuarios, etiquetado de variantes, clasificación de nuevas variantes, y clasificación de variantes de pacientes.

[4]Unified Modeling Language

A continuación se describirá el detalle de los casos de uso a implementar.

Registro y login de usuario

El usuario debe registrarse en el sistema para poder acceder a las distintas funcionalidades, brindando sus datos. Una vez registrado, basta con iniciar sesión para acceder a las operaciones principales de la plataforma.

Entrenamiento de usuarios

Como fue mencionado previamente se busca que los usuarios intenten clasificar variantes de las cuales ya se conoce su etiqueta real. Para clasificar las variantes el usuario debe contar con la información adecuada y ordenada a modo de guía, con una fuerte base en las reglas ACMG/AMP [44]), a partir de las cuales se asume que el usuario realiza la clasificación. Una vez que el usuario aporta su etiqueta se compara la respuesta del usuario con los valores presentes en el *ground truth* del sistema. Finalmente se le brinda retroalimentación al usuario por cada clasificación que emita, la cual se emite en una forma en la que pueda generar un impacto representativo el aprendizaje de los usuarios. Luego de una determinada cantidad de iteraciones, y de acuerdo a la precisión de sus respuestas, se va modificando la reputación del usuario, la cual debe ser sostenida e incrementada de forma activa. En este caso de uso los datos a mostrar consisten en la información asociada a la variante en cuestión: efecto, localización, genes asociados, frecuencia poblacional, función de la proteína asociada, fenotipos relevantes, estudios clínicos asociados, entre otros. Además, se pretende que los usuarios cuenten con un fuerte insumo en cuanto a cómo orientarse al momento de enfrentarse a la clasificación de variantes, a través de las reglas ACMG. Los datos que se reciben consisten en la etiqueta del usuario asignada a la variante, si se trata de un acierto o desacierto, y datos accesorios tales como la fecha, hora de clasificación, entre otros.

Etiquetado de variantes

El otro gran objetivo de la plataforma es el etiquetado de variantes, a modo de aportar a la clasificación de las mismas mediante redes de expertos, a través del aporte de datos y etiquetas a la clasificación mediante aprendizaje automático. En este caso, al usuario se le presentan variantes cuya clasificación es desconocida o con conflictos en su interpretación. Para cada variante presentada al usuario, el mismo debe proponer una clasificación asociada al nivel de patogenicidad. Una vez que el usuario aporta su clasificación, la misma se almacena. La cantidad de etiquetados que realice el usuario en conjunto con la reputación que ha adquirido en su entrenamiento, se toma como insumo para actualizar dicha reputación de forma activa. Generalmente los usuarios que han generado una mayor reputación en la plataforma, y que traigan una mayor destreza de base, son los que más aportarían en el etiquetado de variantes conflictivas, siendo la etapa más avanzada del sistema de niveles de usuarios. En este caso, los datos a presentar al usuario son los mismos que se mencionaron en el caso de uso anterior, con la diferencia de que se habilita de forma explícita la información de las reglas ACMG/AMP [44] que cumple la variante, y no se guía al usuario con retroalimentación a modo de aprendizaje.

Servicio de clasificación de variantes nuevas

Los expertos autenticados y que usen la plataforma participan de forma activa en el aporte de datos para generar métodos de clasificación automática de variantes. Estos métodos, no buscan sustituir el veredicto de los expertos, sino complementarlo o asistirlo. En este sentido, el sistema puede ofrecer a usuarios invitados, o usuarios de la plataforma en general un veredicto sobre la patogenicidad de variantes de interés particular. Esta posibilidad de servicio de clasificación puede venir tanto de los expertos colaboradores en la plataforma, como de la clasificación automática. Los usuarios pueden

subir sus variantes al sistema y obtener el veredicto de un experto respecto a la patogenicidad de la variante, además de poder usar la clasificación basada en aprendizaje automático.

Clasificación de variantes de pacientes

Los usuarios del sistema que cuenten con los permisos requeridos pueden acceder a variantes de pacientes (que no se encuentran en bases de datos públicas) de interés para un grupo de estudio. Si bien el objetivo de la plataforma no consiste en brindar las funcionalidades para almacenar y gestionar información de pacientes, se implementaron las funcionalidades básicas a los efectos de que los expertos avanzados puedan amalgamar tareas de clasificación y evaluación de variantes para diagnóstico.

4.1.6. Modelado de dominio

El modelo de datos se centra en la clasificación de variantes a través de la asignación de una etiqueta de patogenicidad. Estas variantes, dependiendo del tipo al que pertenezcan, contienen una serie de atributos que contribuyen a que el usuario pueda obtener un veredicto sobre su patogenicidad (Figura 4.2). Como fue mencionado anteriormente, dependiendo del caso de uso, los atributos o las relaciones de la variantes con otros elementos del modelo permiten diferenciarlas en distintos tipos: variantes comunes tomadas de bases de datos públicas, variantes internas de la plataforma (brindadas por los usuarios) o variantes de pacientes de interés. Cada usuario puede aportar una etiqueta distinta sobre una misma variante, a partir de las cuales se genera un consenso único para cada variante. Los roles de los usuarios pueden diferenciarse, tanto por su nivel de pericia como los niveles de acceso que contenga. Aquí los niveles de acceso son los que diferencian a los usuarios en cuanto a los datos a los que pueden acceder y las acciones que pueden desarrollar en la plataforma. El consenso se generará a partir de la ponderación de varios aspectos, como las etiquetas brindadas y el nivel de cada usuario.

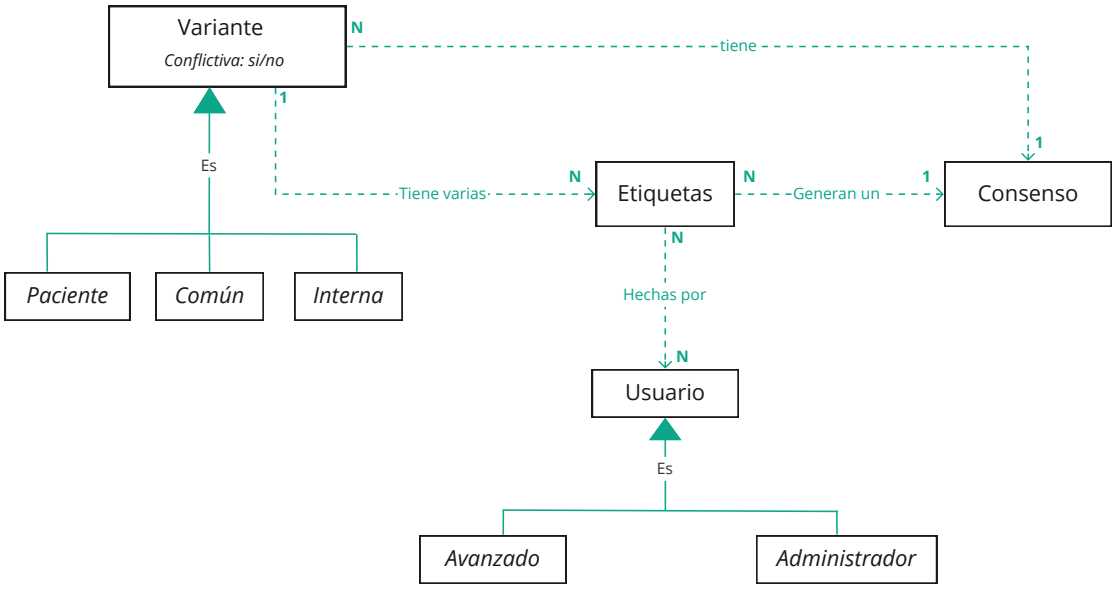


FIGURA 4.2: Modelo de dominio de la plataforma desde el punto de vista de los datos. Las variantes almacenadas pueden ser de distintos tipos, y en lo que refiere a cómo su diferenciación afecta a la funcionalidad de la plataforma, pueden ser: variantes comunes provenientes de bases de datos públicas, de pacientes, o ingresadas por los usuarios de la plataforma (internas). Además, las variantes pueden ser conflictivas o no, lo que puede determinar el caso de uso en el cual aportarán. Las variantes ton etiquetadas por distintos usuarios, y pueden recibir varias etiquetas de los mismos. Aún así, cada variante tendrá asociado una etiqueta consenso única, generada a partir de las etiquetas recibidas.

4.2. Diseño del Sistema Web

4.2.1. Arquitectura

La arquitectura de la versión preliminar del sistema web, representada en la Figura 4.3, principalmente consta de cuatro partes. Por un lado en el *front-end*, el cliente que interactúa con los usuarios a través de la interfaz en un sitio web. Por otro lado, el *back-end* o el servidor en donde se ubica la estructura, lógica del sistema y los datos, el cual contiene: una API, la base de datos, y los modelos predictivos que se integran a la plataforma. La aplicación web se construirá usando tanto un *front-end* como un *back-end* desarrollados en Python. Este último, incluyendo a la base de datos estará alojado en servidores pertenecientes al grupo de Ingeniería Biológica del CENUR Litoral Norte (UdelaR). Como se puede observar en la Figura 4.3, la aplicación Dash hará peticiones HTTP a la API Flask, que a su vez interactuará con la base de datos MySQL, escribiendo o leyendo registros en ella, o con el modelo predictivo sirviéndolo para inferencias en tiempo real.

Los criterios seguidos para la elección de cada tecnología, las decisiones de diseño de la arquitectura, y la descripción de las mismas serán explicados en la Sección 4.2.2, y en la sección actual sólo serán mencionadas las principales herramientas usadas, a los efectos de brindar un primer abordaje a su interacción en la arquitectura.

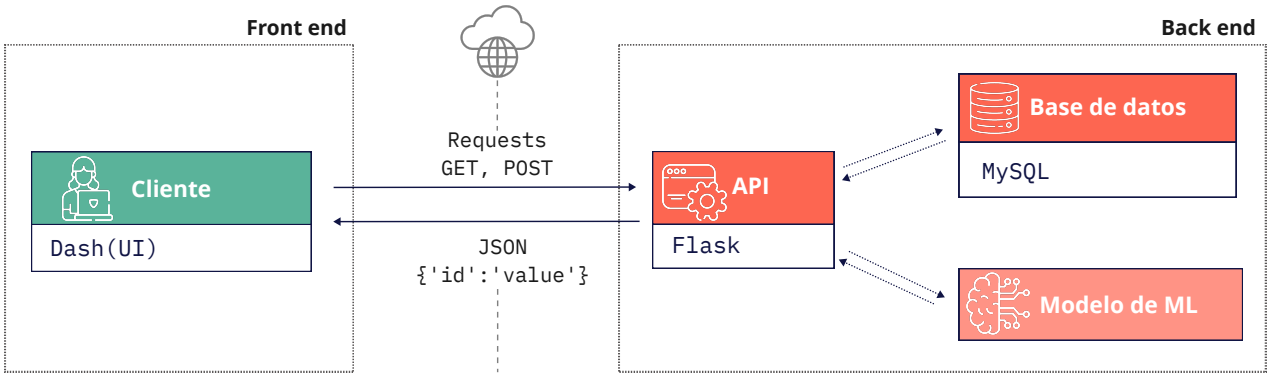


FIGURA 4.3: Diagrama de la arquitectura de la aplicación web a generar. El proyecto está organizado en dos grandes partes: la aplicación web en el *front-end* y el servidor de la aplicación desde el *backend*. La aplicación web interactúa directamente con los usuarios a través de una interfaz generada usando el *framework* Dash [159]. A través de la interacción con la interfaz el usuario hace peticiones al servidor API Flask [160] con servicios web, que a su vez interactuará con persistencia con la base de datos MySQL [161] escribiendo o leyendo registros en ella. Además, se integra a la aplicación el modelo de *Machine Learning* implementado a la plataforma, el cual interactúa con la interfaz y la base de datos mediante la API, sirviendo para inferencia en tiempo real.

A continuación de detalla respecto a los elementos principales de la arquitectura del sistema a implementar:

- ***Backend***:
  - Base de datos: consiste en un esquema relacional, que usa MySQL como manejador. Dentro de esta capa el servidor almacena toda la información referida a variantes, usuarios, clasificación, entre otros. El objetivo principal de esta parte es almacenar y organizar todos los datos que necesitará la aplicación web para su funcionamiento. Los datos almacenados son adquiridos tanto desde la plataforma en si, como de otros servicios y bases de datos públicas, tales como ClinVar. En ambos casos, los datos son procesados previo a su acorde

almacenamiento. Dada la velocidad de actualización de las bases de datos externas que sirven de insumo, esta base de datos se actualiza como mínimo semanalmente.

- API: Se cuenta con una API REST implementada en Flask [160], que será la encargada de recibir las peticiones desde el *frontend* y realizar las consultas necesarias a la base de datos. También será la encargada de unir los resultados y disponibilizarlos de forma clara y ordenada en el *frontend*. El protocolo de comunicación entre los módulos es HTTPS. El propósito principal de la API es recuperar los datos necesarios para para brindarle funcionalidad a la plataforma, tanto en su interacción con el usuario, como con los modelos. Esta capa es potencialmente la más crítica de la arquitectura, ya que la recuperación de datos adecuada es fundamental para el propósito de la aplicación. Además de que contar con una API le brinda independencia a las distintas partes del sistema, haciendo que no sean dependiente la lógica de la visualización. Esto será fundamental dada la instancia preliminar del proyecto.

- ***Frontend***: esta capa de funcionamiento es capaz de utilizar los datos recuperados a través de la API para renderizar la página web. Esta página web será la interfaz a través de la cual el usuario final acceder al sistema, y allí podrá realizar principalmente 2 tareas operativas desde el punto de vista de objetivo de la plataforma: el etiquetado de variantes nuevas y la clasificación guiada de variantes conocidas. El *frontend* implica una app multi-página implementada utilizando Dash, adaptando la modalidad de *dashboard* original a los aspectos visuales requeridos por el usuario final.

El detalle del desarrollo e implementación del modelo de *machine learning* implementado para la clasificación de variantes será detallado en el próximo capítulo.

4.2.2. Elección de tecnologías

Para la implementación del sistema web se analizaron diferentes tecnologías previamente a la selección de las mencionadas en la sección anterior, tanto para el desarrollo del *frontend* como del *backend*. Las mismas fueron analizadas desde un punto de vista de funcionalidad, aplicabilidad y adaptación como solución propuesta. La elección de tecnologías se realizó partiendo de las condiciones propuestas sobre cada parte. En este sentido, el almacenamiento de los datos tendría que ser implementado a los efectos de obtener un MVP, y la interfaz gráfica debería ser una interfaz de prototipado, alcanzando la metodología de PoC.

4.2.2.1. Lenguaje de programación

La elección del lenguaje de programación quedó sujeta a considerar el que mejor se adapte a cada gran parte del sistema general (*frontend*, *backend*, API). Que la elección se adapte a cada instancia no significa que se haya buscado la mejor opción para su logrado desde un punto de vista tecnológico, sino que se buscaron los medios que se adapten a los requerimientos mínimos establecidos. A pesar de ello, se considera acorde, que dada la experiencia previa generada en un lenguaje multipropósito como lo es Python, puede ser interesante profundizar en el mismo para objetivos distintos.

4.2.2.2. Front-end

La condición de este proyecto respecto a la interfaz de usuario, es que la misma debería consistir en una interfaz de prototipado, más allá de obtener un producto final competitivo en el mercado. Esto se fundamenta en que al obtener las funcionalidades principales completas en una primera instancia, ya es suficiente para cumplir con los objetivos funcionales principales de la plataforma.
El principal criterio para la elección de tecnologías a usarse en el desarrollo de la plataforma web en general implica la reducción de tiempo de desarrollo y la simplicidad en la solución. Esto permitiría

que el trabajo en cada instancia se centre en la lógica de la aplicación. Para ello, es que se optó por la utilización de *frameworks* de *dashboarding* para la implementación del *frontend*, que permitieran la generación de una web, sin la necesidad de tener experiencia o la manipulación de herramientas de desarrollo web. Estas herramientas de *dashboarding*, además de brindar una interfaz visualización de datos donde se podría resumir fácilmente la información de las variantes a estudiar, permiten manipular y resumir datos, interactuar mediante la entrada de usuarios, y atender a las solicitudes de éstos mediante el uso de servidores web. Si bien cabe destacar que el objetivo ideal de la plataforma no es que sea un *dashboard*, sino una aplicación web genérica desde su base, en esta instancia inicial se prioriza la facilidad en el desarrollo, además de las herramientas de análisis de datos que puedan brindar los *frameworks* mencionados.
En la exploración de distintos *frameworks* de código abierto para construir interfaces de visualización de datos, se identificó una gran variedad de alternativas, tales como Shiny [162], Dash [159], Streamlit [163], Voilà [164], Panel [165], entre otras. Cada herramienta, con enfoques y fortalezas distintos.

Al momento de la elección de la tecnología para la interfaz web, se realizaron pruebas iniciales a los efectos de conocer de forma preliminar y así descartar entre tres *frameworks* seleccionados a partir de los mencionados, teniendo en cuenta su creciente popularidad y características: Shiny, Streamlit y Dash (Figura 4.4). Las primeras dos implementaciones llegaron a instancias únicamente de exploración y prueba, a distintos grados de funcionalidades parciales, pero sin lograr una implementación completa.

La primera herramienta explorada fue Shiny [166], un paquete de R lanzado en 2012, que permite generar aplicaciones web interactivas, y que ha tenido un amplio uso en bioinformática en los últimos años [167]. Esta herramienta, se comporta como el resto de las otras evaluadas, en el sentido de que genera todo el código necesario para la creación de una aplicación web, sin necesidad de entender en detalle el funcionamiento de las tecnologías web [166]. Shiny se exploró contemplando su potencialidad, estabilidad y tiempo de desarrollo, amplia gama de aplicación, además de la fortaleza del grupo de trabajo en la manipulación de R. En este caso se implementó una modificación de ejemplos pre-elaborados por Shiny para mostrar un conjunto de datos de forma gráfica, e interactuar con dicho gráfico a través de la modificación de parámetros componentes de la interfaz. Esta primera implementación permitió comprender la estructura de una aplicación Shiny, el flujo de información dentro de la misma, y cómo desarrollar y publicarla en la web, quedando a disposición en un repositorio de GitLab para futuras primeras experiencias y posibles ampliaciones [168]. Finalmente, la elección del lenguaje de programación, y las limitaciones impuestas por la versión gratuita de la herramienta, llevaron a que la prueba se cierre en esta instancia. Respecto al lenguaje de programación, posterior a las instancias de selección de tecnologías se lanzó una versión de Shiny para Python [169]. Si bien a la actualidad se encuentra en una versión experimental, queda evaluada su posibilidad de uso para trabajos futuros.

Considerando lo definido previamente respecto al lenguaje de programación a utilizar, se tendió a implementar la plataforma en un *framework* cuyo lenguaje central sea Python, pero que a su vez pueda extenderse a otros lenguajes usados comúnmente en el área de la Bioinformática como R. Antes de converger a Dash, se realizaron pruebas de implementación en Streamlit, una librería de código abierto desarrollada en 2019, que también está teniendo un amplio uso para desarollo de interfaces web en bioinformática [170] [171] [172]. En este caso, se implementaron casos de ejemplo al igual que en Shiny, y además se realizó una prueba con algunos componentes más que las pruebas anteriores. Esta prueba implicó: el manejo de un menú mediante el cual se pueda acceder a distintas páginas, la implementación de un acercamiento a una aplicación multipágina y su navegación, la interacción con la interfaz a través de del ingreso de datos mediante elementos de la interfaz y archivos, un abordaje inicial a la selección de etiquetas para variantes cargadas mediante archivos externos, y la conexión e interacción con una base de datos mediante MySQL. Todas estas estas pruebas también se

encuentran depositadas en un repositorio del proyecto global [168]. Streamlit permite la generación de aplicaciones web interactivas simples con facilidad y con una curva de aprendizaje moderada, sin embargo, a medida que aumenta la complejidad y la intensidad de su uso, las dificultades comienzan a evidenciarse. En este caso, si bien la implementación final de esta prueba fue directa y rápida, la herramienta resultó restrictiva desde varios aspectos: al momento de la prueba la opción de app multipágina no era una de las características funcionales principales de Steamlit; la autenticación de usuarios no podía implementarse por fuera de la versión en la nube o a través de terceros; en términos de escalabilidad, Streamlit no tiene un buen desempeño en el uso simultáneo por varios usuarios, ni con volúmenes de datos grandes; en términos de interactividad, Streamlit resulta poco flexible .

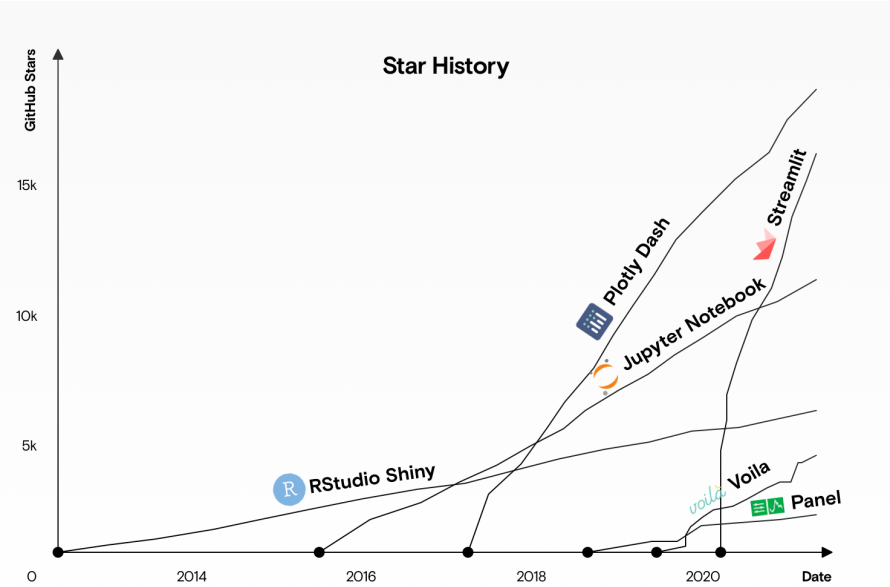


FIGURA 4.4: Comparación de popularidad de *frameworks* de *dashboarding* entre 2012 y la actualidad, medida en la cantidad de estrellas en los repositorios Github de los proyectos. En los últimos años, Streamlit y Dash han tenido un aumento en su popularidad como soluciones integrales en el desarrollo de aplicaciones web. Extraído de [173].

Teniendo en cuenta las experiencias realizadas previamente (con los criterios preliminares evaluados resumidos en la Tabla 4.1), se procedió a evaluar Dash, herramienta con la cual se trabajará. Además del criterio mencionado respecto a lenguaje de programación, Dash es el único que soporta Python, R, Julia y F# (experimental) [159] y presenta cualidades que resultaron interesantes para el presente proyecto:

- Soporta aplicaciones multi-página, lo que resulta de relevancia en la aplicación web deseada. A pesar de que el intercambio de variables entre las páginas tiene que hacerse explícitamente, lo que puede obstaculizar la aplicabilidad, la aplicación a realizar no requeriría que las páginas compartan mucha información con frecuencia.
- Presenta flexibilidad en el diseño.
- Permite la autenticación de usuarios, lo que es un requisito primordial en un sistema que usa como insumo los datos de los mismos.
- Presenta facilidad en el mantenimiento de aplicaciones.
- Está bien documentado.

- Presenta una popularidad y soporte creciente de la comunidad (Figura 4.4). A la fecha es el que más popularidad presenta *online* principalmente a nivel de foros.
- Permite desarrollar una aplicación web ocultando la complejidad de escribir en herramientas de desarrollo web de cero, además de permitir presentar y manipular datos en un *dashboard*.
- Es "reactivo", permitiendo la implementación de interfaces de usuario complejas con múltiples entradas, salidas y entradas que a su vez dependen de otras.
- Las aplicaciones que permite desarrollar son intrínsecamente multiusuario: varios usuarios pueden ver las aplicaciones y tener sesiones independientes.
- Es frecuentemente usado en plataformas que integran modelos predictivos sobre datos en tiempo real.

El resumen de los criterios evaluados en esta versión preliminar de la plataforma para la elección de la herramienta de *dashboarding* se presenta en la Tabla 4.1. Los criterios considerados en este caso fueron:

- Madurez: evaluada en función de la estabilidad y antigüedad del proyecto.
- Popularidad: basada en la recepción de la herramienta y la cantidad de estrellas de GitHub.
- Soporte: evaluado en función del soporte de la comunidad principalmente en foros.
- Simplicidad: de acuerdo a qué tan fácil que es empezar a usar la librería.
- Flexibilidad/adaptabilidad: basada en lo flexible que es la librería.
- Escalabilidad: evaluada en función de cómo se adaptan las aplicaciones al aumento en la complejidad de las mismas y a la demanda dada el aumento de usuarios manipulando las aplicaciones en simultáneo.
- Soporte multi-página: basado en si la herramienta brinda formas de implementar aplicaciones multi-página, y qué tan directa es esta implementación.
- Flexibilidad en versión libre: referida a qué tan flexible y qué tantas herramientas se brindan para el desarrollo sin la necesidad de utilizar una versión privativa.
- Interactividad: basada en la flexibilidad para que el usuario interaccione con los datos a través de la interfaz.
- Arquitectura *back-end*: define el tipo de arquitectura utilizada por la API en cada herramienta.
- *Front-end*: define la tecnología que usa cada *framework* para el desarrollo del *front-end*.
- Lenguaje: respecto a los principales lenguajes de programación que soporta la biblioteca.

Criterio	Dash	Streamlit	Shiny
Madurez	Media	Baja	Alta
Popularidad	Alta	Alta	Media
Soporte	Alto	Medio	Alto
Simplicidad	Media	Alta	Media
Flexibilidad/adaptabilidad	Media	Baja	Media
Escalabilidad	Alta	Baja	Media
Soporte de estructura multi-página	Alto	Bajo	Medio
Flexibilidad en versión libre	Media	Alta	Media
Interactividad	Alta	Baja	Media
Arquitectura back-end	Stateless	Stateful	Stateful
Front-end	React	React	jQuery
Lenguaje	Python, R, Julia, F#	Python	R

CUADRO 4.1: Comparativa entre tres*frameworks* de *dashboarding* para la generación *low-code* de interfaces web, según los criterios de interés para el presente trabajo. La comparación se hace entre los *frameworks* que fueron evaluadas de forma empírica: Dash, Shiny y Streamlit [174] [172] [175, 173]. El código de colores indica qué tanto se cumple cada criterio en cada tipo de DBMS: verde en caso de cumplir con el criterio de forma óptima siendo apto para el proyecto, amarillo en caso de cumplir con el criterio de forma aceptable, pero que puede implicar algunas dificultades en el desarrollo del proyecto, y rojo cuando el criterio no se cumple, no siendo apto para el proyecto

4.2.2.2.1. Dash

Como fue mencionado anteriormente, la herramienta elegida para el desarrollo del *front-end* es Dash.

Dash es una librería *open source* de Python, desarrollada por Plotly Technologies Inc. [159], lanzada oficialmente de forma pública en el año 2017. El objetivo del *framework* es la construcción de aplicaciones web interactivas que permitan a los usuarios explorar y visualizar datos de una manera fácil de usar. Si bien Dash es una herramienta de libre acceso bajo licencia MIT, también se ofrece un complemento privativo (*Dash Enterprise*), el cual proporciona servicios específicos a empresas [176].

Las principales tecnologías utilizadas por Dash, representados en la Figura 4.5 son:

- Flask: es un *framework* de desarrollo web de Python que es utilizado por Dash como servidor *back-end* para manejar la lógica del lado del servidor y las interacciones con la base de datos [160].
- React: es una librería JavaScript [177] para la construcción de interfaces de usuario. Es utilizada por Dash para construir el **front-end** de la aplicación web, proporcionando una interfaz de usuario responsiva y dinámica [178].
- Plotly: es una potente librería de visualización de datos que permite a los usuarios crear tablas y gráficos interactivos. Es compatible con varios tipos de gráficos [179].
- Plotly.js: es la biblioteca JavaScript que potencia Plotly. Se utiliza para crear visualizaciones de alta calidad como tablas y gráficos interactivos y con capacidad de respuesta en el *front-end* de la aplicación web. Plotly.js está construido sobre D3.js (para la exportación de imágenes vectorizadas con calidad de publicación) y WebGL (para visualización de alto rendimiento) [180].

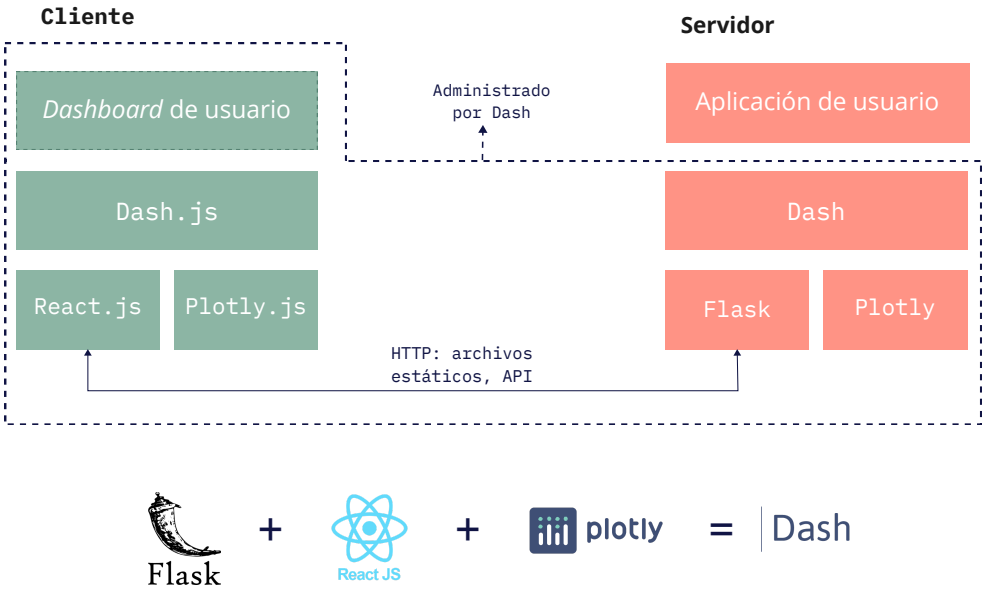


FIGURA 4.5: Arquitectura en capas general de una aplicación Dash, con sus componentes principales. Dash está compuesto por: Flask como servidor *back-end*, Plotly como generador de gráficos, y React para la manipulación de componentes interactivos en el *front-end*. Extraído y modificado de [181] y [182]

Además de estos componentes principales, Dash se vale de varios paquetes, para su funcionamiento, entre los que se encuentran: Dash, el paquete principal que sostiene toda la aplicación; *Dash Core Components* (DCC) [183], que brinda componentes interactivos a ser manipulados por los usuarios como desplegables, *sliders*, selectores, entre otros; *Dash HTML Components*, que proporciona todas las etiquetas de HTML para que estén dispobibles como clases de Python [184]; *Dash Bootstrap Components* (DBC), un paquete que añade funcionalidad Bootstrap a Dash [185], encargándose del diseño y señales visuales codificadas; entre otros [181].

En resumen, Dash es una poderosa herramienta para la construcción de aplicaciones web interactivas, con interfaces sencillas y potentes, mediante el uso de tecnologías y bibliotecas de gran uso en la actualidad, disponibles para proporcionar una experiencia fluida e intuitiva tanto para los desarrolladores como para los usuarios finales. Las aplicaciones Dash son de por si servidores web que ejecutan Flask y comunican paquetes JSON a través de peticiones HTTP. El frontend de Dash renderiza los componentes utilizando React.js [181].

La gran potencialidad de Dash radica en la capacidad de crear aplicaciones web analíticas con HTML [186] y CSS [187] dentro de Python, además de manejar la interactividad de la página sin la necesidad de escribir código de JavaScript. Una vez que el usuario implementa el código Python, Dash no solo general el código en Javascript necesario, sino también la API web para crear y actualizar el contenido de la aplicación web en el navegador [159] [181] . Es decir, que desde el punto de vista del desarrollador, no es necesario conocer desarrollo web o cómo se integran los componentes mencionados previamente, sino que basta con manejar un lenguaje (Python en este caso) para crear tanto el *front-end* como el *back-end* de una aplicación web. Esto disminuye la complejidad y la demanda de tiempo al momento de implementar una plataforma con los requerimientos mínimos esperados.

Lo mencionado anteriormente es especialmente útil para quienes se desempeñan en ciencia de datos, dado que el manejo de la aplicación creada se realiza usando código Python puro, y que la herramienta hace que sea muy sencillo construir una interfaz gráfica de usuario en torno a un

código de análisis de datos para visualizar e interactuar con grandes cantidades de datos. Para este fin Dash incluye componentes como Graph que representa gráficos con plotly.js. Plotly.js, el cual es declarativo, de código abierto, rápido y admite una amplia gama de gráficos científicos, financieros y empresariales [159].

En relación a la aplicación puntual que se le plantea dar a la herramienta, Dash cuenta con un desarrollo interesante en el ámbito de la Bioinformática, fundamentado por la demanda de esta área, la manipulación de gráficos interactivos y la interacción con grandes cantidades de datos. Dash Bio [188] es un conjunto de componentes bioinformáticos específicos adicionales, como la visualización de alineamientos de secuencias y árboles filogenéticos interactivos, que simplifican el análisis y la visualización de datos biológicos complejos y la interacción con los mismos en una aplicación Dash [189].

Aunque Dash es una potente herramienta para crear aplicaciones web, existen algunas desventajas y dificultades a tener en cuenta a la hora de desarrollar con esta herramienta, y que fueron claves a ser consideradas en este proyecto (Tabla 4.1):

- Dash es relativamente nuevo, lo que significa que pueden haber recursos o elementos que aún estén en fase de desarrollo. Más allá de que tenga un alto soporte en la comunidad, puede que la misma no cuente con las herramientas para cubrir estas faltas.
- Puede resultar difícil de personalizar la estructura subyacente de la aplicación y su comportamiento. Esto puede hacer que sea también difícil integrar Dash con otras herramientas y tecnologías, limitando la flexibilidad de la aplicación.
- Las aplicaciones Dash pueden ser más complejas de desplegar que las aplicaciones web tradicionales debido a la necesidad de ejecutar un servidor independiente, lo que requiere conocimientos de servidores web y desarrollo.
- Dash tiene un enfoque al mercado empresarial y no incluye todas sus funciones disponibles en la versión de código abierto, por lo que pueden haber limitaciones al momento de querer explorarlas en proyectos como el actual.
- Expone mucho más de HTML, CSS y Javascript subyacente para el usuario que las otras herramientas evaluadas. Es por ello que es frecuente que el desarrollo y mantenimiento de aplicaciones desarrolladas en Dash requieran de cierta experiencia o entendimiento en el desarrollo *front-end*.

Como ya fue mencionado, Dash está construido sobre Flask. Esto significa que no sería necesario implementar una API, y se podría crear una plataforma completa mediante código de Dash únicamente. Sin embargo, como se expresó en la Sección 4.2.1, la arquitectura de la plataforma a desarrollar en el presente trabajo separa la API de Dash. Todo esto se hace por una razón: **independizar la lógica de la visualización**. Que la interfaz para la implementación actual alcance requerimientos mínimos implica que a futuro la misma pueda tener cambios importantes, lo que incluye reemplazar la aplicación Dash con otra tecnología *front-end*, o añadir una aplicación de escritorio o incluso móvil. Por ello, en este caso Dash se utiliza únicamente para el *front-end*.

4.2.2.3. Backend

4.2.2.3.1. Base de datos

El uso de una base de datos en la aplicación cumple con múltiples propósitos, los cuales se irán detallando más adelante en este y otros capítulos. Para el almacenamiento centralizado de los datos se partió desde la posibilidad de elección de distintos tipos de sistemas de gestión de bases de datos.

De esta forma, al momento de decidir la tecnología para la gestión de la base de datos se va a discutir entre tipos de base de datos, y no entre manejadores específicos.

La integración de grandes conjuntos de datos biológicos, especialmente en el campo de la genómica, es una una tarea multidisciplinar que implica un hábil manejo en tanto en aspectos de ciencias de la vida como en el campo de la informática. El sistema general necesita un motor de integración y unificación de los datos a almacenar. El sistema de almacenamiento de los datos debe ser capaz de satisfacer las exigencias altas de las aplicaciones provenientes de la genómica con enfoque clínico, como la anotación de genomas, exomas y paneles de genes. La plataforma debe proporcionar el acceso a cerca de 100 conjuntos de datos genómicos por variante tomados de distintas bases de datos públicas, lo que representa un volumen de datos elevado, además de las contribuciones de la comunidad que se piensa obtener. La base de datos debe además contemplar la actualización con nuevas variantes, y que cada vez que se actualiza una base de datos pública, el sistema debe procesar la información rápidamente y ponerla a disposición nuevamente en un plazo breve. La calidad del manejo de los datos es de suma importancia, ya que se debe garantizar que los mismos se integren de forma meticulosa, cuidando que las nuevas inserciones coincidan de forma consistente en todos los recursos de datos disponibles en la plataforma. Además debe de contemplarse la realización de controles exhaustivos de la integridad de los datos.

Las tecnologías utilizadas para generar datos de variantes y su anotación ha evolucionado rápidamente en los últimos años, y una cuestión clave es cómo almacenar de forma eficaz dichos datos y, al mismo tiempo, poner la información a disposición de los usuarios. Históricamente, las bases de datos relacionales han proporcionado gran parte del *framework* para el almacenamiento y la recuperación de datos, ya que para datos estructurados ofrecen una mayor eficiencia de almacenamiento y velocidades de recuperación de datos en comparación con los archivos planos [190, 191]. Sin embargo, a medida que ha aumentado la tasa de generación y adquisición de datos, han surgido nuevos enfoques para el almacenamiento de los mismos, los cuales han permitido contribuir en el almacenamiento de datos no estructurados y semi-estructurados, con los cuales los sistemas relacionales son menos eficientes [192, 191]. Así los sistemas de bases de datos distribuidos denominados *Not Only SQL* (NoSQL) han permitido el almacenamiento y procesamiento eficiente de datos masivos [193, 194, 192].

Los esquemas NoSQL han estado siendo explorados en la gestión de conjunto de datos biológicos y clínicos de distintas naturalezas, y han identificado beneficios sobre el uso de este tipo de bases de datos [195, 191, 196]. En lo que respecta a la aplicación en el almacenamiento, recuperación y análisis de variantes genómicas, se han explorado ambos enfoques [197, 198, 199] [200, 201, 202, 203]. Los sistemas NoSQL representan un grupo de herramientas interesantes para almacenar y recuperar conjuntos de datos caracterizados por su alto volumen, variabilidad y velocidad con la que se generan [204, 205, 206]. En este sentido, las variantes genómicas pueden incluirse en esta categoría. Tanto el volumen de datos como la velocidad se atribuyen al elevado ritmo al que se generan las variantes gracias a las nuevas tecnologías de secuenciación, cada vez más rápidas y de alto rendimiento [46]. Por otro lado, las variantes son un conjunto de datos muy variable de por si, dado los diversos tipos que pueden existir, sus efectos y asociaciones. Al momento de almacenar este tipo de datos, se han realizado grandes esfuerzos para lograr la estandarización en el guardado y la presentación de variantes y sus características asociadas, como el formato VCF ([71]) mencionado anteriormente. Sin embargo, las variantes deben traer anotaciones para poder asistir, en este caso, su clasificación, y la misma es difícil de estandarizar. Dado que varias características de esta anotación no son ajustables a todas o todos los tipos de variantes o la aplicación que se le quiere dar, es común que se cuente con muchos datos faltantes. En un contexto de datos estructurados, este aspecto mencionado anteriormente llevaría a esquemas ineficientes, dada la escasez de información [200]. Como consecuencia, las bases de datos relacionales estructuradas se han dejado de considerarse como la mejor opción para tratar con variantes genómicas con anotación heterogénea. Muchos proyectos han optado por

esquemas no relacionales, los cuales además han mostrado un mejor rendimiento que las bases de datos relacionales tanto en términos de escalabilidad horizontal como de tiempo computacional en la recuperación de datos [200, 191, 206]. Para este objetivo se han explorado DBMS como MongoDB, Cassandra, HBase y CouchDB [207, 195, 191, 200]. No obstante, existe un equilibrio entre la flexibilidad del modelo de datos y la complejidad de las consultas: las bases de datos NoSQL no suelen disponer de un lenguaje de consulta similar al SQL, y las consultas deben calcularse previamente para construir las correspondientes estructuras de índices en memoria que permitan realizar búsquedas rápidas. Más allá de la potencialidad detrás de las bases de datos NoSQL, ambos esquemas se usan en la actualidad [203] para el almacenamiento de variantes y su correspondiente anotación.

Criterios para la selección del tipo de base de datos.

Previo a la decisión de qué tipo de base de datos usar en el proyecto actual, se tomaron determinados criterios para su evaluación. Los criterios evaluados se tuvieron en cuenta en base a las necesidades del proyecto, y las características del grupo de trabajo. Los criterios a tener en cuenta fueron:

- Experiencia: se considera un factor importante la utilización de un manejador del cual se cuente con una mínima experiencia previa, no sólo por los tiempos de desarrollo, sino también pon el conocimiento en general de las funcionalidades que brinda cada tecnología.
- Flexibilidad en las consultas: se requiere contar con la posibilidad de realizar consultas que brinden una variedad de opciones y parámetros a definir.
- Rendimiento: al tratarse de una base de datos que potencialmente sea de gran porte en el futuro, es necesario analizar qué gestor es más veloz en los distintos tipos de operaciones.
- Escalabilidad: como se mencionó en el ítem anterior, se considera que el volumen de datos puede aumentar de forma significativa en un futuro, por lo que se esperaría que la tecnología adaptarte a dicho aumento.
- Integridad: se procura que el DBMS garantice la integridad de los datos almacenados y procesados, y su usabilidad en todo momento.
- Facilidad de instalación: la plataforma debe poder funcionar en cualquier sistema con mínimas capacidades de recursos.
- Seguridad: dada la naturaleza de los datos, es fundamental que el sistema ofrezca seguridad y madurez en distintos niveles de control de acceso a los datos.

Con el objetivo de hacer un breve análisis sobre las características generales de algunos manejadores de bases de datos, en la Tabla 4.2 se expone un análisis comparativos respecto a 7 criterios entre sistemas de administración de bases de datos (DBMS[5]) relacional y no relacional.

Como se ha expuesto anteriormente, el uso de DBMS no relacionales ha cobrado importancia en el manejo de datos genómicos en el último tiempo, específicamente en variantes y su anotación. Si a lo anterior agregamos los criterios expuestos en la Tabla 4.2, el balance puede desplazarse tanto por lo relacional como no relacional. No obstante, se optó en este trabajo por utilizar un esquema relacional, específicamente MySQL como DBMS [161]. Para la instancia del proyecto global en este trabajo de tesis basta con la elaboración de una base de datos de tipo relacional, lo que incluso, dadas las características de la obtención datos, resulta el abordaje más natural. Anteriormente se planteó que el abordaje de la plataforma debería ser la obtención de un MVC; aún así, la intención es que

[5]DataBase Management System, en inglés

Criterio	DBMS relacional	DBMS no relacional
Experiencia	Alta	Baja
Flexibilidad en consultas	Alta	Intermedia
Performance de consultas	Alta	Alta
Performance de inserción/actualización	Alta	Alta
Escalabilidad	Baja	Alta
Integridad	Alta	Intermedia
Facilidad en instalación	Alta	Alta
Seguridad	Alta	Intermedia

CUADRO 4.2: Comparativa entre tipos de DBMS relacional y no relacional, según ocho criterios considerados de relevancia [206]. El código de colores indica qué tanto se cumple cada criterio en cada tipo de DBMS: verde en caso de cumplir con el criterio de forma óptima siendo apto para el proyecto, amarillo en caso de cumplir con el criterio de forma aceptable, pero que puede implicar algunas dificultades en el desarrollo del proyecto, y rojo cuando el criterio no se cumple, no siendo apto para el proyecto.

el *back-end* sea planteada de una forma más robusta que el *front-end*. Cabe destacar, que más allá de que la decisión del DBMS alcance requerimientos mínimos, no se encuentra alejado de abordajes actuales en el estado del arte [203]. Igualmente, se considerará como trabajo futuro la exploración de DBMS de tipo no relacional, con un énfasis en bases de datos orientadas a documentos [191].

MySQL es un DBMS que se caracteriza por su alto rendimiento al realizar distinto tipo de operaciones, su facilidad de configuración e instalación, su soporte en una amplia variedad de sistemas operativos, su capacidad de mantener la integridad de los datos y su bajo costo en requerimientos para la elaboración de bases de datos [161]. Esto último significa un bajo consumo de procesamiento que permite ser ejecutada en una máquina de escasos recursos. Utiliza lenguaje SQL el cual permite realizar operaciones muy complejas sobre la base de datos, lo que resultará fundamental a los objetivos del presente proyecto.

4.2.2.3.2. API

En la Sección 4.2.2.2 se adelantó que más allá de que Dash ofrece una solución integral desde el desarrollo tanto del *front-end* como del *back end*, la arquitectura del sistema incluye una API separada de Dash, a los efectos de independizar el *back-end* de la interfaz. Además de esto, el uso de una API se justifica a través de las siguientes ventajas [208]:

- Flexibilidad: el manejo de las APIs es simple y se encuentra estandarizado. Los datos no están ligados a métodos, por lo que no es necesaria su adaptación a un formato particular, además de que se pueden manejar múltiples tipos de llamadas. Esta flexibilidad permite el desarrollo de APIs para una gran variedad de aplicaciones, las cuales pueden ser muy diversas en sus objetivos.
- Compatibilidad: una API funciona como una interfaz estándar que puede comunicarse con diferentes tipos de *frontend*, abarcando desde aplicaciones móviles hasta aplicaciones web. Mientras el *frontend* envíe la misma petición a la API, obtendrá el mismo resultado de vuelta independientemente de la naturaleza de dicha interfaz, por lo que una API como estándar permite la interoperabilidad entre muchos sistemas diferentes.
- Seguridad: uno de los propósitos fundamentales de las APIs es ocultar los detalles internos del funcionamiento de un sistema a través del encapsulamiento de la lógica interna del mismo. De esta forma se exponen sólo aquellas partes que se consideran útiles. Con esta disposición,

no se da acceso directo a la base de datos, por ejemplo, la cual puede contener información información sensible. La información interna no será revelada hacia el exterior de la aplicación.

- Independencia y portabilidad: debido a la separación entre el cliente y el servidor, el protocolo facilita que los desarrollos de las diferentes partes de un proyecto se puedan dar de manera separada, permitiendo a los componentes evolucionar independientemente unos de otros. Además, siempre y cuando se cumpla con el requisito de que los datos de cada petición sean enviados de forma correcta, es posible que el *frontend* y el *backend* se pueden alojar en servidores diferentes. En este caso, en el cual el proyecto no tenía fijo inicialmente el sitio en el que se alojaría la aplicación web, significaba una gran ventaja.
- Escalabilidad y balance de carga: gracias a la separación entre el cliente y el servidor, el producto puede crecer en complejidad sin que ello represente dificultades mayores.

Además de que la herramienta elegida para la implementación del *frontend* se basa naturalmente en Flask, se ha tendido a la utilización de dicho *framework* para el desarrollo de la API. Existen principalmente tres tipos de *frameworks* para Python en el desarrollo de APIs: *full-stack*, *microframework* y asíncrono. Flask en particular, es un *microframework* que que puede ser utilizado para construir una aplicación web de forma fácil, rápida, confiable y escalable. Si bien un *framework* de aplicación web como Flask proporciona los paquetes necesarios y módulos que hacen el trabajo pesado por el desarrollador, no ofrece funcionalidades y características adicionales, tales como la capa de abstracción de la base de datos. De allí, el "micro"de *micro-framework*: Flask no toma todas las decisiones por el usuario, permitiendo extender la base de funcionalidades que ofrece según las necesidades de la aplicación ([208]).
También existen otros *frameworks* web disponibles en el mercado basados en Python y que pueden interactuar con aplicaciones basadas en Dash. Un fuerte competidor de Flask en este sentido es Django, un *full-stack framework*. Si bien Django es una herramienta muy potente, para los objetivos del trabajo es suficiente con contar con una herramienta minimalista como Flask.

4.3. Implementación

La plataforma a desarrollar, denominada inicialmente **actG-Learn**[6] aspira a ser una herramienta web gratuita y de código abierto, tal como fue descrito anteriormente. A continuación serán descritos los detalles de su implementación. Primero se presentará cómo es el flujo de usuario dentro de la plataforma, lo que dará pie a cómo el usuario lo realiza desde la interfaz realizada. Posteriormente se brindarán algunos detalles de la implementación del *back-end*, en lo que refiere a decisiones de diseños relevantes o diferenciales para el presente trabajo.

4.3.1. Flujo de usuario

Primero serán descritos los componentes principales de la plataforma desde un punto de vista funcional, para luego explicar a modo general cómo se implementaron dichas funcionalidades en cada aspecto, tanto desde la interfaz, como desde el *back-end*. La implementación del sistema se llevó adelante con el objetivo de que se cumplan las siguientes funcionalidades desde el punto de vista del usuario final de la plataforma:

- Permita la autenticación del usuario.

[6] `actg`: por el acrónimo para los cuatro tipos de bases nitrogenadas que se encuentran en la molécula de ADN. *learn*: por la presencia de la palabra "aprendizaje" a lo largo del proyecto (aprendizaje de usuarios y uso de herramientas de aprendizaje automático). `G-learn`: por la el objetivo a futuro de contar con una herramienta que use estrategias de juego en el aprendizaje de la clasificación de variantes.

- Permita una instancia de clasificación de variantes a modo de entrenamiento del usuario.
- Permita el etiquetado de variantes nuevas como insumo para el sistema total.

Inicialmente se planteó como caso de uso que la plataforma sirva al usuario como servicio de clasificación. Este caso de uso se encuentra brindado, pero no con una funcionalidad completa, por lo que no será desarrollada como caso de uso principal en la implementación actual. Cabe destacar que las dos ultimas funcionalidades listadas anteriormente cumplen con un orden sugerido. Si bien el fin principal de la plataforma implica el etiquetado de variantes como insumo para la clasificación automática, el insumo del nivel de experto es requerido, y esto se obtiene a partir de la funcionalidad de entrenamiento de usuario. Además, esta última funcionalidad resulta un beneficio al usuario en cuanto a ganar experiencia clasificando variantes. En este sentido, si se establece el flujo en etapas, se insta a que el usuario primero se entrene, y luego etiquete. La implementación de la plataforma busca reflejar estos aspectos.
El flujo del usuario se describe de forma general en la Figura 4.6. El usuario accede a la plataforma en la web, y lo primero que se le ofrece al ingresar es un espacio de autenticación. Una vez identificado el usuario, el mismo procede a elegir entre los dos casos de uso principales: clasificar variantes a los efectos de ganar experiencia y puntaje en el sistema, o pasar a etiquetar variantes con conflicto de etiqueta como aporte al sistema de clasificación automática. En una implementación completa, podría ingresar una variante propia a la cual se le aplique el algoritmo de clasificación logrado.

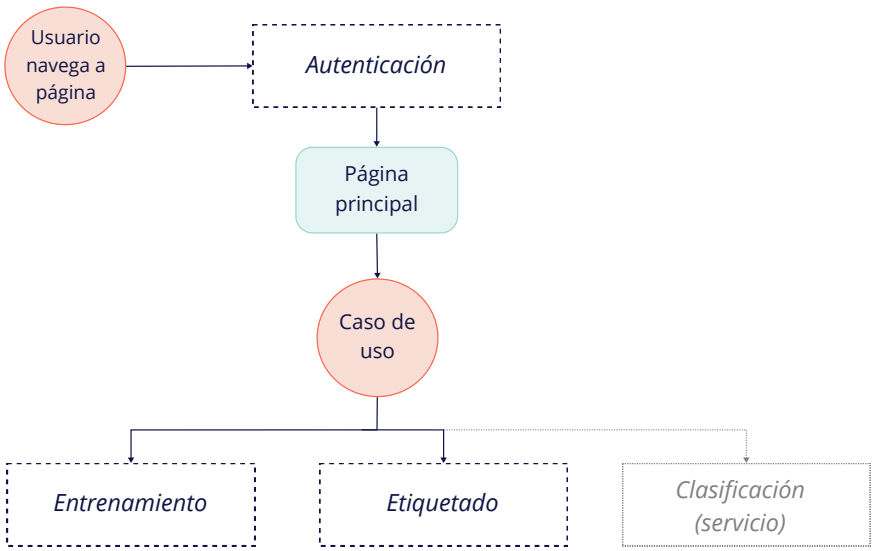


Figura 4.6: Diagrama de flujo general de la navegación de los usuarios en la aplicación web. El usuario ingresa a la página web, y lo primero que se encuentra es con la etapa de autenticación. Una vez completada, ingresa a la página principal, la cual conecta al usuario con los dos casos de uso posibles y completos en su implementación: por un lado la clasificación de variantes a modo de entrenamiento del usuario, y por otro lado el etiquetado. Los procesos representados en líneas punteadas serán desarrollados a continuación. En cuanto al servicio de clasificación de variantes, se encuentra en una instancia inicial de implementación, por ello se encuentra en color claro.

Cada gran bloque funcional representado en la Figura 4.6, opera como se muestra en la Figura 4.7, en la cual desarrolla cada instancia de forma detallada.

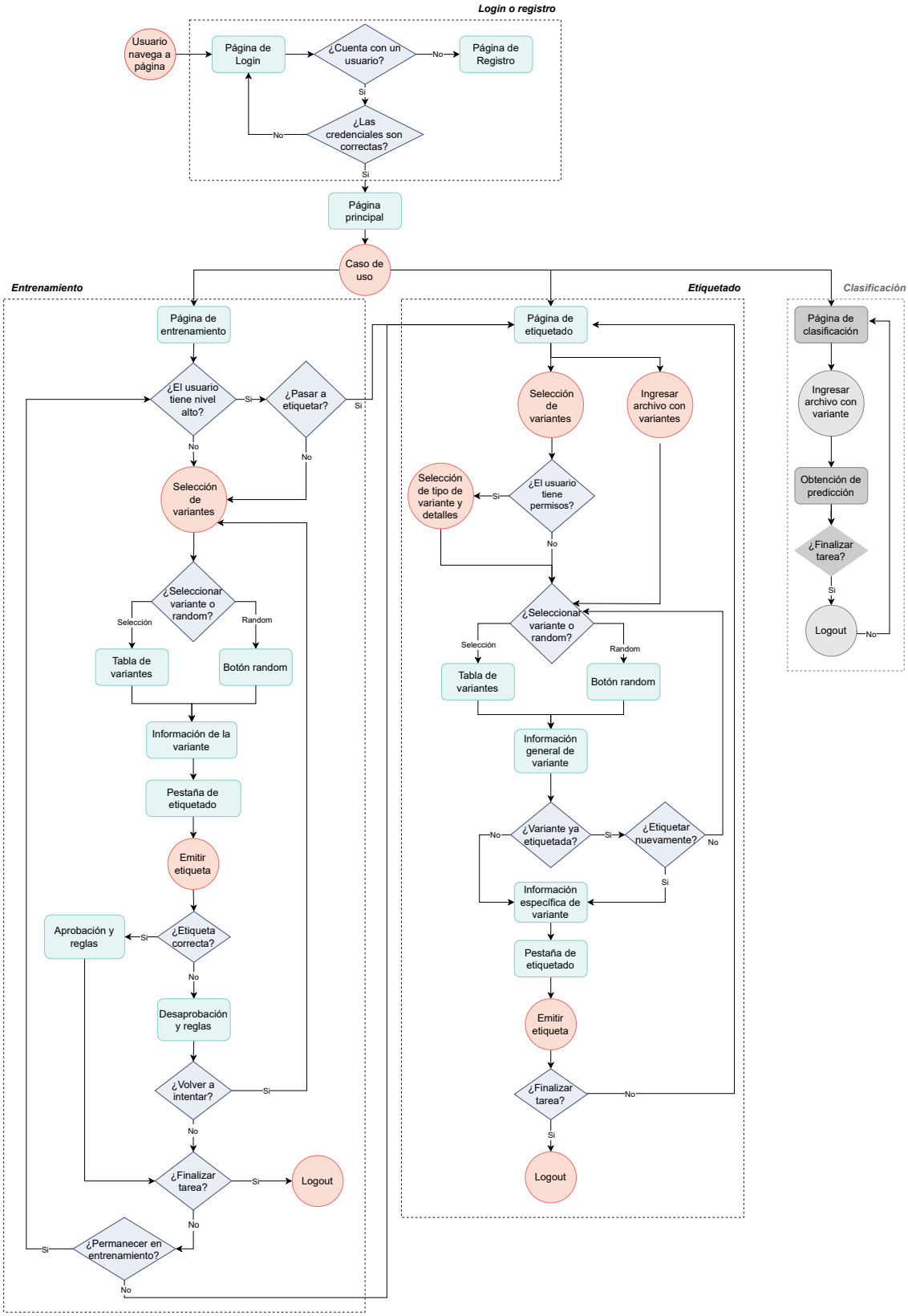


FIGURA 4.7: Diagrama de flujo detallado de la navegación del usuario. Al ingresar a la plataforma, el usuario puede autenticarse con sus datos, o registrarse. Una vez dentro, en el caso de uso de entrenamiento o aprendizaje se puede seleccionar o elegir una variante aleatoria para emitir una etiqueta, y en función de la correctitud de la misma, se brinda un *feedback* y actualizan las estadísticas del usuario. En caso de seguir el proceso de etiquetado, se indica el modo de selección de variante y se genera una etiqueta según la información brindada, la cual se almacena para posterior generación de un consenso. La instancia de clasificación actualmente implica brindar una predicción de la patogenicidad de una variante ingresada a través de un archivo.

Como se muestra en la Figura 4.7, una vez que el usuario ingresa desde la web, se le solicita el registro en la misma. Si el usuario ya está registrado en la plataforma, solamente acede a colocar sus datos (nombre de usuario y contraseña), los cuales serán solicitados hasta que se trate de los datos correctos. Si el usuario no se encuentra registrado en la plataforma, entonces se accede a dar la opción de registro; una vez que cuenta con un nombre de usuario y contraseña el procedimiento es igual al mencionado anteriormente para un usuario ya registrado. Una vez autenticado al usuario se despliega la página inicial. A partir de allí el usuario no tiene restricciones de acceso a las dos principales funcionalidades de la plataforma, sin embargo, en función de su elección, se le pueden realizar sugerencias. Cabe destacar, que algunas funciones de la implementación, actualmente no se encuentran disponibles en la web, a los efectos de mantener la versión de desarrollo bajo control. En este sentido, el registro de nuevos usuarios en la plataforma cumple con esta restricción. Si bien se encuentra implementado como parte de la autenticación, actualmente no está disponible al público, y los usuarios que participan en la plataforma, ya cuentan con sus credenciales.

Una de las opciones de funcionalidad a elegir, tomado también como la primera etapa en el flujo de trabajo, el usuario puede acceder a la página de entrenamiento de usuarios, destinada a la clasificación de variantes desde un punto de vista de aprendizaje del usuario. Al acceder a esta página se evalúa el nivel de experiencia con el que cuenta el usuario. En esta instancia al usuario se le presentan variantes que ya cuentan con una etiqueta con un alto grado de consenso, las cuales tiene que lograr etiquetar correctamente de acuerdo a los datos brindados. Esto aporta al usuario, y al sistema una valoración del nivel de experiencia del usuario en cuanto a la determinación de patogenicidad de variantes. Si éste cuenta con un nivel elevado, más allá de que la clasificación en esta instancia le permitirá mantener dicho nivel, se le sugerirá aportar la clasificación de variantes conflictivas, brindando insumos al sistema de clasificación. Ante la sugerencia, el usuario puede optar por migrar a la página de etiquetado, o continuar clasificando variantes conocidas. En el último caso, el usuario pasará a decidir cómo quiere obtener las variantes a clasificar: puede realizar una selección en una tabla, o puede solicitar una variante aleatoria del sistema. En ambos casos, el despliegue de la información de la variante se realiza de la misma forma: una primera vista con la información general de la variante, y luego una disposición en forma de pestaña detalla en cada una de ellas la información según su naturaleza. Si el usuario navega, llega finalmente a la pestaña donde elabora la clasificación de la variante, aportándole una etiqueta. Al emitirla, el sistema le ofrece un *feedback*: si la etiqueta es la correcta respecto a un consenso, se brinda un mensaje de aprobación y las reglas que lo fundamentan; si la etiqueta es incorrecta, se detallan las razones por las cuales la variante no cumple con las características de la etiqueta asignada. En ambos caso, internamente se actualizan las estadísticas del usuario. En este último caso, se sugiere repetir la tarea de clasificación, la cual se puede hacer sobre la misma variante (siempre y cuando se haya elegido de la tabla). Más allá de esta sugerencia que se realiza únicamente para quien ha clasificado de forma incorrecta, el usuario puede continuar clasificando variantes en general, cambiar de tarea, o salir del sistema.

Un segundo camino luego de la página principal es pasar directamente al etiquetado de variantes. A diferencia de la página de clasificación, no se hace ninguna sugerencia inicial respecto al nivel de usuario. En este caso, lo ideal es que el usuario brinde etiquetas a variantes que no tengan un consenso cuando éste ya cuente con un nivel de experiencia elevado, a los efectos de que el aporte sea lo más seguro posible. Sin embargo, si el usuario aún es principiante, el sistema de clasificación automática debe tener en cuenta el nivel de experto al momento de asignar una etiqueta, a través de un nivel de confianza Esto permite que se le asigne un peso a la etiqueta de acuerdo a la experiencia del usuario, por lo que no se considera necesario advertir al respecto. De hecho, dado que se requieren muchas etiquetas y el proceso de adquisición es en general lento, se tiende a no establecer restricciones de ningún tipo en esta instancia. Una vez que se accede a la página principal de etiquetado, se hace la selección de la variante sobre la cual se quiere trabajar. En este caso agrega un paso a la selección de la variante respecto al procedimiento explicado anteriormente. Este paso implica que, para usuarios

que tengan permisos especiales, se le presente variantes de distinto tipo para su etiquetado. Esto puede incluir variantes de pacientes, o generadas con fines de prueba, por lo que el usuario que las manipule tiene que estar estrechamente relacionado con el desarrollo de la plataforma, en un principio. En esta instancia, el usuario si cuenta con permisos decide si quiere trabajar con variantes especiales o estándar, para luego pasar a indicar el tipo de proceso de selección de la variante: aleatorio o selección en una tabla. Tras la selección la variante, e independientemente del mecanismo elegido, se presenta información general de la variante en donde, en parte, se indica si la misma ya fue etiquetada previamente. El usuario puede elegir entre permanecer, y aportar más etiquetas a la variante, o elegir una nueva. De la misma manera que en la instancia anterior, el usuario navega la información específica de a variante hasta emitir una etiqueta. En esta etapa no se le brinda un *feedback* detallado al usuario respecto a la clasificación, dado que es una instancia solo de recepción de etiquetas, y no se cuenta con un consenso para las variantes analizadas que sea a externo al sistema (*ground truth*). Luego del etiquetado el usuario puede optar por etiquetar una nueva variante, o salir del sistema.

4.3.1.1. Proceso común de clasificación de variantes

El flujo de trabajo planteado anteriormente mantiene implícito un proceso que es transversal, tanto al entrenamiento de usuarios como para el proceso de etiquetado de variantes. Esto implica que la herramienta guíe a los usuarios a través de un proceso de curación de variantes a través del cual se puede determinar el impacto de las mismas. Este proceso puede tener variaciones, pero en general se basa en el uso de las reglas ACMG/AMP, las cuales aportan rigor y calidad en la clasificación de variantes de la línea germinal. Como fue descrito anteriormente, estas reglas describen guías estandarizadas para la clasificación de variantes en 5 categorías: patogénica, probablemente patogénica, benigna, probablemente benigna y de significado incierto. Para obtener un consenso, el personal dedicado a la clasificación de variantes se basa en distintas características de la misma: características funcionales, frecuencia, predicciones *in sílico* ya realizadas, evidencia aportada por otros médicos, entre otras. Este proceso general de evaluación puede observarse en la Figura 4.8.

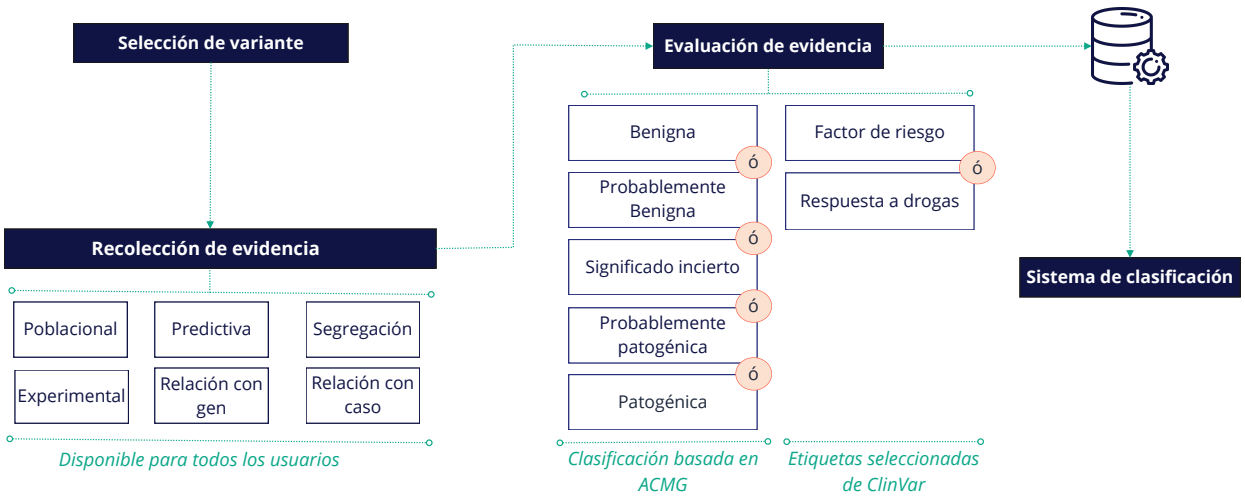


FIGURA 4.8: Resumen del proceso de curación de variantes. Los usuarios reciben una variante y evalúan las características de las mismas que entran en seis categorías. Con esta evidencia el usuario realiza una clasificación final de patogenicidad de acuerdo con las regla ACMG/AMP. Luego este insumo se almacena en la base de datos de atcG-Learn.

Si bien el proceso de curación aplicado en actG-Learn sigue lineamientos establecidos en reglas ya estandarizadas, el mismo fue elaborado teniendo en cuenta el flujo particular seguido por profesionales en el área locales. Atendiendo a este flujo particular, a las categorías de clasificación establecidas por ACMG/AMP, se le han incluido otras usadas en este caso por ClinVar. Específicamente, y para la evaluación posterior de su comportamiento, se han incluido etiquetas como factor de riesgo y respuesta a drogas.

4.3.1.2. Datos

Los principales datos que brindan la base funcional a la plataforma, en función de los cuales se construyen las demás estructuras y funcionalidades son las variantes. Éstas serán analizadas para su clasificación, ya sea con el objetivo de aprendizaje como de etiquetado. Los tipos de variantes se pueden definir en función de varios aspectos (efecto, localización, nivel de patogenicidad, etc.), pero en lo que compete a la sección actual, desde el punto de vista funcional de la plataforma, se identifican 3 tipos de variantes que pueden ser presentadas al usuario dependiendo de la tarea ejecutada (Figura 4.9):

- Variantes comunes o por defecto: provienen de repositorios públicos, tales como ClinVar o 1000 Genomas. Dependiendo de la tarea a ejecutar, este conjunto de variantes puede cambiar.
- Variantes internas: son ingresadas por los usuarios de la plataforma.
- Variantes de pacientes[7]: son conjuntos de datos pertenecen a muestras de estudio (en general en proceso de diagnóstico), y que son brindados por usuarios colaboradores de la plataforma, o que lo solicitan explícitamente. Cabe destacar que estos datos sos sensibles y no puede hacerse públicos en general, por lo cual pueden ser accedidos sólo por usuarios específicos.

4.3.1.2.1. Datos en aprendizaje/entrenamiento

Se utilizan variantes por defecto provenientes de ClinVar y 1000 Genomas. En base al objetivo de esta instancia, el conjunto de datos que está detrás contiene variantes con una clasificación ya definida previamente, con un alto nivel de aserción y evidencia que lo soporte.
En este sentido, en el caso de los datos de ClinVar, se busca que las etiquetas no impliquen conflicto, y en general se mantienen las categorías propuestas por las reglas ACMG/AMP. Esto último acompañado de un *review status* en las categorías "*Practice guideline*" (cuatro estrellas) o "*Reviewed by expert panel*" (tres estrellas).
Acompañando el proceso de aprendizaje, los conjuntos de variantes disponibles pueden variar en cada instancia de clasificación, dependiendo del usuario al frente: a medida que el nivel del usuario aumenta, el conjunto de variantes que se le asignan van aumentando en complejidad también. Los detalles de esta implementación se describen más adelante.

4.3.1.2.2. Datos en etiquetado.

En esta instancia se usan los tres tipos de variantes: por defecto, internas y de pacientes. Teniendo en cuenta el objetivo de desambiguar conflictos en etiquetas, el aspecto común entre estos tres tipos de datos es que su clasificación sea conflictiva (según la definición establecida previamente) o que no cuenten con una clasificación previa.
En el caso de los datos de ClinVar, se considera el subconjunto de variantes que contiene la categoría

[7] *La funcionalidad de análisis de datos de pacientes con un fin diagnóstico no está implementada en la plataforma, dado que no se considera entre los objetivos del trabajo, sin embargo estos datos fueron ingresados con el fin de acelerar el proceso de desarrollo y testeo.*

"*Conflicting interpretations of pathogenicity*" incluida en el significado clínico, ya sea individualmente, como combinada con otras categorías.

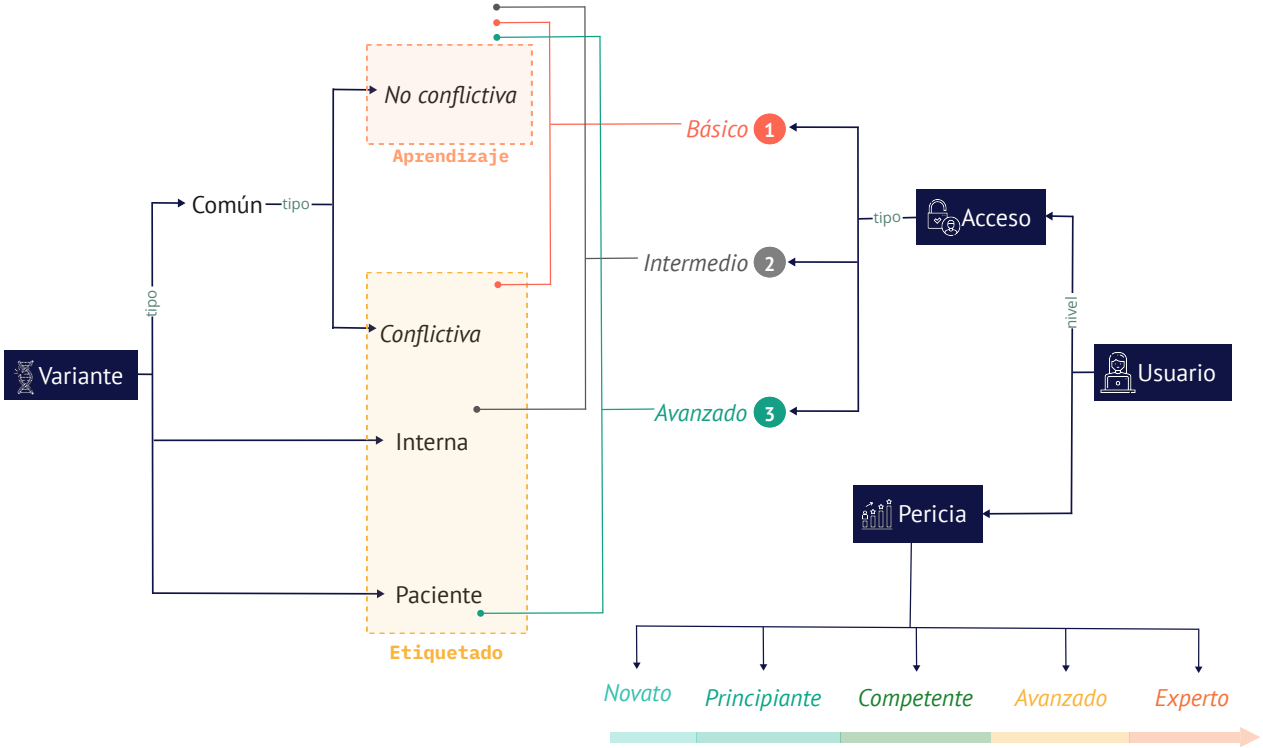


FIGURA 4.9: Diagrama de interacción entre los tipos de variantes y usuarios según su funcionalidad en la plataforma. Desde este punto de vista se distinguen tres tipos de variantes: comunes o por defecto (de bases de datos públicas), internas a la plataforma (brindadas por usuarios) y de pacientes. Las variantes comunes a su vez pueden ser conflictivas o no conflictivas (según el grado de consenso en su etiqueta). Los tipos de variantes se dividen según el caso de uso: la tarea de aprendizaje usa variantes comunes no conflictivas, mientras que el resto son destinadas al etiquetado. Los usuarios pueden clasificarse según el nivel de acceso a variantes y su nivel de pericia. En el primer caso depende de las variantes a las que pueden acceder: un usuario avanzado puede acceder a todas las variantes, el intermedio solo a las internas y comunes, y el básico solo a las comunes. Por el nivel de pericia los usuarios pueden pertenecer a cinco categorías; de menor a mayor nivel: novato, principiante, competente, avanzado y experto.

4.3.1.3. Tipos de usuarios

Al momento de presentar los casos de uso y el modelo de dominio en la Sección 4.1, se planteó que los usuarios de la plataforma pueden clasificarse según dos criterios: nivel de acceso y de pericia. En la Figura 4.9 se puede observar la categorización y las interacciones que define a dicha categorización. Según el nivel de acceso hay tres tipos ya definidos. El nivel de acceso más bajo (usuario nivel 1), implica que de variantes obtenidas de repositorios públicos. En este nivel se encuentra el usuario estándar pero también el invitado que solo accede al servicio de clasificación. El nivel de acceso intermedio (usuario nivel 2) permite el análisis de variantes internas a la plataforma compartidas por otros usuarios. El nivel de acceso superior (usuario nivel 3) permite el análisis de datos sensibles y protegidos.

El nivel de acceso más elevado no está contemplado en procesos automatizados que lo asignen (por ejemplo, que considere el nivel de pericia o la institución del usuario), sino que se asigna a demanda por el usuario administrador únicamente (que por defecto está incluido la categoría).

Además del tipo de usuario según los permisos con los que cuenta, se consideran niveles de pericia,

los cuales se consideran como reguladores tanto del modelo de aprendizaje como de la clasificación automática de variantes. En este sentido, los niveles de usuario considerados son:

- Novato o principiante: el usuario no es familiar al proceso de clasificación de variantes, nunca estuvo en contacto con las reglas ni conoce las características asociadas a las variantes.
- Principiante avanzado: el usuario conoce los lineamientos a seguir al momento de clasificar variantes, pero nunca ha llevado a la práctica la tarea con datos reales, y es muy probable que necesite asistencia continua para poder realizar la tarea.
- Competente: comprende bien las reglas de clasificación de variantes, y ha logrado realizar el proceso de clasificación en instancias previas. Aún necesita asistencia para realizar la tarea, y reafirmar conceptos.
- Avanzado: puede clasificar variantes con fluidez de forma desatendida, y tiene una buena comprensión de los aspectos conceptuales detrás de la tarea, aunque aún no puede formar a otros usuarios y requerirá actualización en el ámbito.
- Experto: el usuario ha alcanzado el nivel máximo en experiencia clasificando variantes, lo que se asocia a aquellos que realizan la tarea a diario como parte de sus actividades en genómica médica, desempeñando el diagnóstico basado en datos de secuenciación de ADN. En general se trata de sujetos que desarrollan, aplican y/o enseñan la competencia con una frecuencia considerable, desde hace un tiempo prolongado. En el caso del grupo de trabajo en el marco del cual se desarrolló la plataforma, los niveles de expertos están ocupados por médicos genetistas cuya tarea principal radica en el diagnóstico mediante herramientas genómicas.

Estos niveles se asignan inicialmente a cada usuario según su propia valoración, sin embargo no son estáticos. La actividad del usuario en la plataforma permite su actualización. Los mecanismos mediante los cuales se actualizan los niveles de pericia serán detallados más adelante.

4.3.2. Interfaz de usuario y navegación

En la sección 4.3.1 se realizó una descripción del flujo de usuario desde un punto de vista funcional, y en la presente sección se presentará el proceso desde el punto de vista de la interfaz de usuario, destacando los aspectos más relevantes del mismo.
El flujo descrito en 4.3.1 es visualizado por el usuario como se muestra en la Figura 4.10, donde se presenta un mapa de navegación del usuario por la herramienta. Como se puede observar, la aplicación consiste en múltiples páginas, dentro de las cuales la información se va desplegando de forma secuencial, y en pestañas. A grandes rasgos, la Figura 4.10 muestra cómo la estructura general se mantiene en cada caso de uso, diferenciándose en tareas puntuales. El detalle de estas diferencias se mencionará en las secciones que siguen.

Al ingresar a la página de actG-Learn el usuario deberá colocar sus credenciales para el inicio de sesión. Una vez iniciada la sesión, se puede comenzar a navegar.El primer sitio al que se dirige al usuario luego de ingresar es a la página de inicio, en la que se presenta una breve descripción de la plataforma, sus objetivos y casos de uso. Desde allí, el usuario puede acceder a los distintos casos de uso, a través de la barra de navegación, la cual es visible en todas las páginas y contiene los enlaces a: la página principal, la página de aprendizaje, página de etiquetado, página de clasificación, página del usuario y al cierre de sesión.

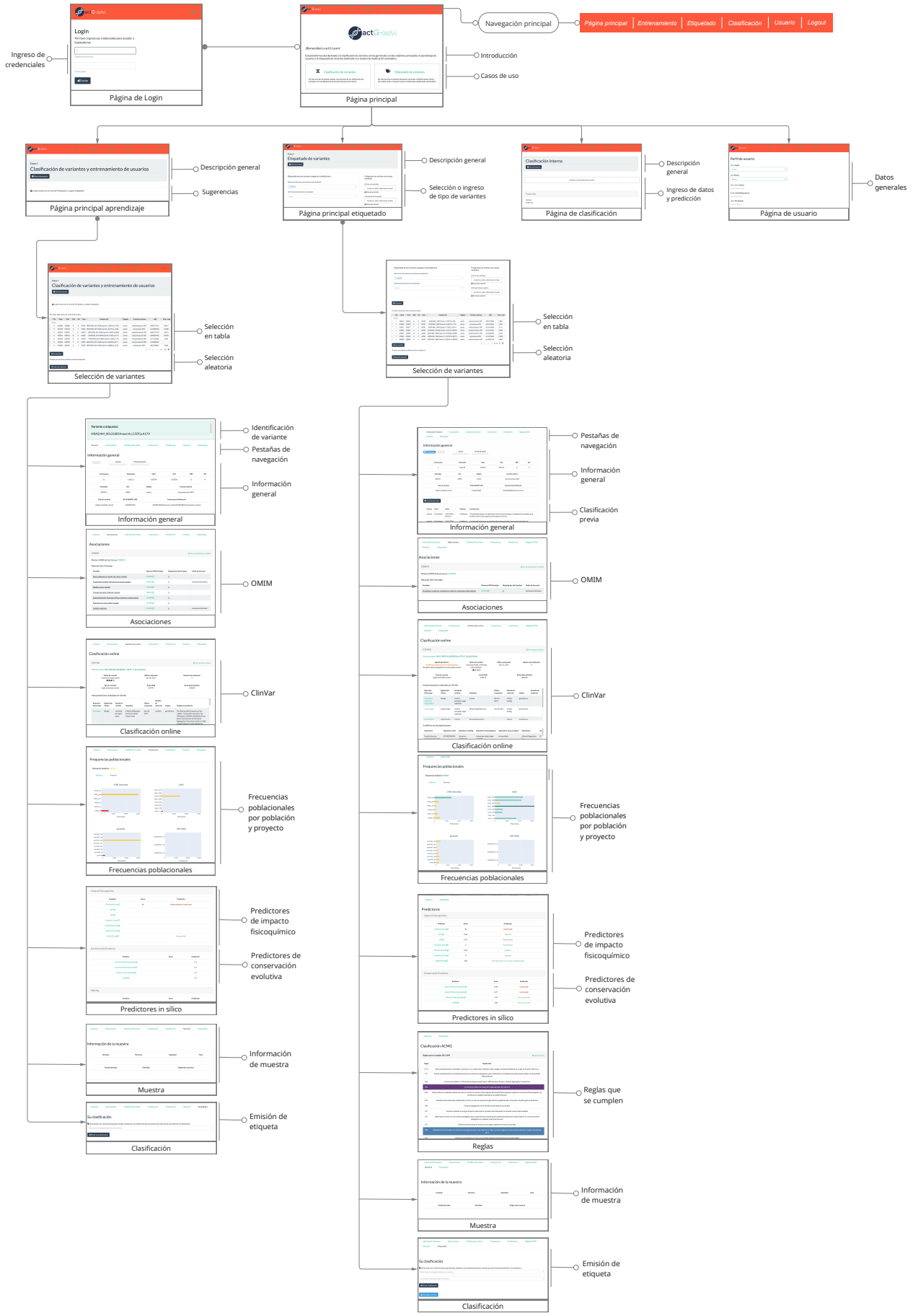


FIGURA 4.10: Mapa de navegación en la interfaz de usuario de la aplicación web.

4.3.2.1. Selección de variante

Cuando aplica el caso de uso más abarcativo (etiquetado) el proceso de selección de variantes implica poder seleccionar aquellas que pertenecen al conjunto de datos de la plataforma, o ingresar variantes nuevas a partir de un archivo. Ambas posibilidades convergen en un proceso único que consiste en dos instancias: selección del tipo de variante a etiquetar, y selección de la metodología de obtención de variantes.
La primera etapa depende de las características del usuario que inició sesión en la plataforma. Si el usuario cuenta con el nivel de acceso más elevado, entonces se pondrán a disposición los 3 tipos de variantes que contiene la plataforma (en cuanto a su origen): las variantes públicas por defecto, las que son internas a la plataforma, y las que pertenecen a pacientes de estudio (desplegado a través de un *dropdown*). Si el usuario cuenta con un nivel de acceso intermedio, entonces las opciones se reducen a variantes internas y por defecto. Si el usuario cuenta con un nivel de acceso mínimo, solamente podrá acceder a las variantes por defecto. Dependiendo del tipo de variante seleccionada, puede ser necesario obtener más detalles. Por ejemplo, en caso de que el usuario con nivel de acceso más elevado seleccione para etiquetar variantes de pacientes, se puede requerir el conjunto de datos para una sola muestra, o para un conjunto de ellas. En este sentido, se habilitará la opción para que se seleccione la muestra de estudio. Esta parte del proceso se ilustra a través de una captura en la Figura 4.11.

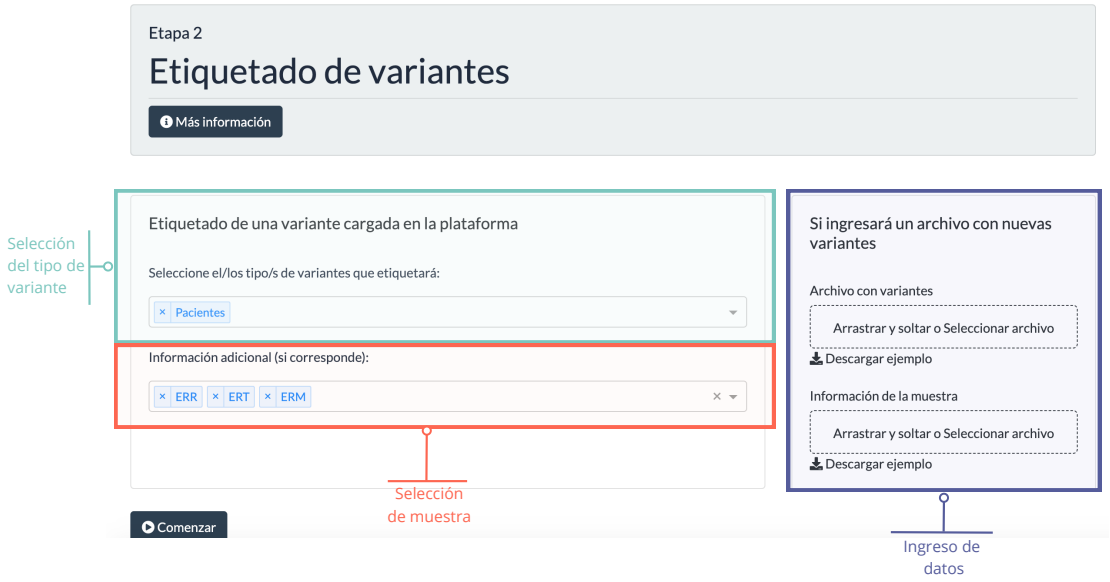


FIGURA 4.11: Selección inicial del tipo de variante a trabajar en el proceso de etiquetado (en recuadro verde) y sus detalles (recuadro naranja). Si el usuario tiene permisos especiales, entonces se le despliegan como opciones los tres tipos de variantes según su origen. En la ilustración se presenta un caso en el que el usuario, al seleccionar datos de paciente puede optar por la muestra de interés. En esta etapa el usuario también puede elegir cargar sus propias variantes como aporte a la base de datos del sistema.

La segunda etapa consiste en elegir el método a través del cual se obtiene la variante a analizar: de selección o aleatorio. En el método de selección el usuario puede elegir la variante a clasificar, a través de su selección en una tabla (Figura 4.12). Esta tabla muestra 8 atributos de que permiten identificar a grandes rasgos una variante y sus características distintivas: cromosoma, posición de inicio, posición final, alelo de referencia, alelo alternativo, cambio aminoacídico producido, región genómica donde se ubica el cambio, función exónica de la variante y símbolo del gen en el que se

encuentra. En la pantalla principal la tabla presenta solo unas pocas variantes, y la misma puede ser recorrida para navegar todo el contenido. Además de navegar por páginas, se puede filtrar el contenido de la tabla con el fin de reducir a variantes de interés, y hacer la selección sobre éstas. En la Figura 4.12 se muestra un ejemplo en el que se filtra la columna de la región, indicando que se buscará sobre un subconjunto de variantes exónicas. Además de la selección directa, el usuario puede elegir una variante de forma aleatoria, a través de un botón que brinda una variante de esta forma desde el conjunto completo de datos.

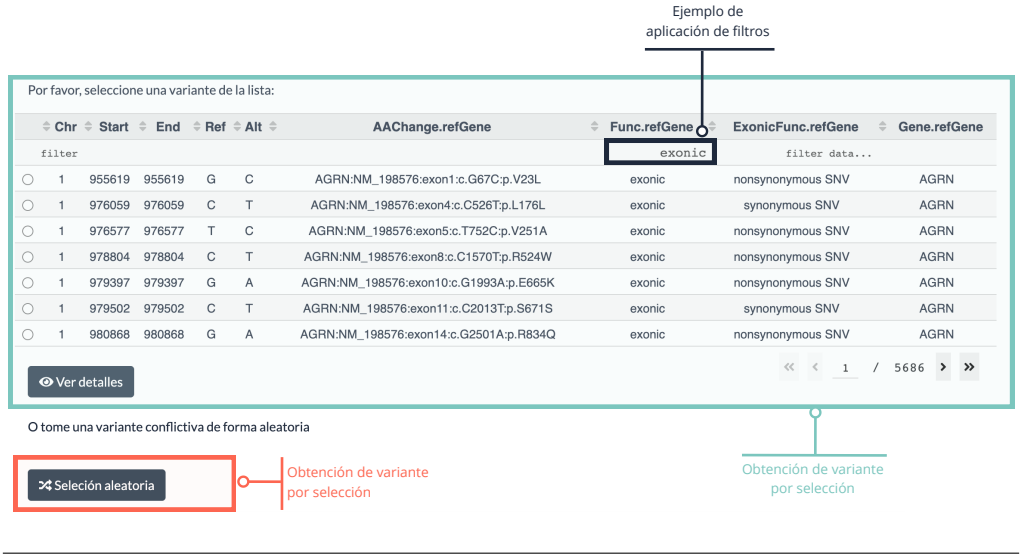


FIGURA 4.12: Selección del método de obtención de una variante para su clasificación, sobre una captura de la plataforma en funcionamiento. En la parte superior (bloque verde) de ilustra cómo se muestran variantes para seleccionar en una tabla; se resalta en azul oscuro un ejemplo de filtro que puede ser aplicado sobre la tabla. En la parte inferior (resaltado en color naranja) se muestra la selección aleatoria.

El proceso general de selección de variante varía cuando se está realizando el uso de la clasificación con fines de entrenamiento de usuario. En este caso, sólo se procede a elegir el método de selección de variantes, ya que se trabaja únicamente con variantes por defecto obtenidas de repositorios públicos. No se despliega la posibilidad de elegir tipos de variantes o sus detalles.

4.3.2.2. Interpretación de la variante y evidencia

Una vez que se obtiene una variante se ofrece una vista en donde se presenta la evidencia relacionada con la variante. En esta vista la interfaz muestra datos de interés de fuentes externas, además de los reportes realizados respecto a la misma por otros usuarios de la plataforma. Los datos mostrados fueron seleccionados en conjunto con médicos genetistas locales, con el fin de seleccionar del conjunto total de características anotadas sobre las variantes, aquellas de relevancia en la aplicación de las reglas. Las características de las variantes mostradas son presentadas en la Tabla 4.3.

Tipo de información	**Datos mostrados**
Información general	Transcriptos RefSeq Citobanda Símbolo del Gen Región de impacto Función exónica ID de variante Tipo de variante Consecuencia molecular
Asociaciones	OMIM: - Fenotipo - Número MIM - ID de mapeo del fenotipo - Modo de herencia
Clasificación Online	ClinVar: - Transcripto primario - ID de alelo - ID NCBI - Interpretación general - Status de revisión - Última interpretación - Número de remitentes - Tipo de variante - Interpretaciones enviadas - Conflictos de interpretación
Frecuencia Poblacional	Genome Aggregation Database (gnomAD) Exome Aggregation Consortium (ExAC) 1000 Genomas Exome Sequencing Project (ESP)
Predictores	Grantham score SIFT LRT Mutation Taster PolyPhen2: - HVAR - HDIV CADD phyloP 100 Way Vertebrate phyloP 20 Way Mammalian SiPhy (29-way) log Odds dbscSNV11 GERP
Reglas	Intervar

Cuadro 4.3: Resumen de la información presentada en la interfaz tomada de fuentes externas. Los datos mostrados para cada variante pueden agruparse según las pestañas mostradas, relacionadas con los tipos de variantes.

La información tomada de fuentes externas en conjunto con la información interna asociada a las variantes y plataforma se despliegan al usuario en forma de siete pestañas, agregando una más para la emisión de una etiqueta/significado clínico. Cada pestaña contiene un tipo de datos distinto, ordenando la información a los efectos de guiar al usuario en el proceso planteado en la Sección

4.3.1.1 (Figura 4.13). El objetivo es que el usuario tenga una experiencia de análisis de variantes que se encuentre basada en las guías ACMG/AMP. Una vez que se ingresa esta instancia, la información se agrupa en: información general, asociaciones, clasificación *online*, frecuencia poblacional, algoritmos predictivos, información del caso y reglas ACMG. La última pestaña no aporta información que aporte a la interpretación, sino que corresponde al sitio donde se ingresa la etiqueta o la clasificación.

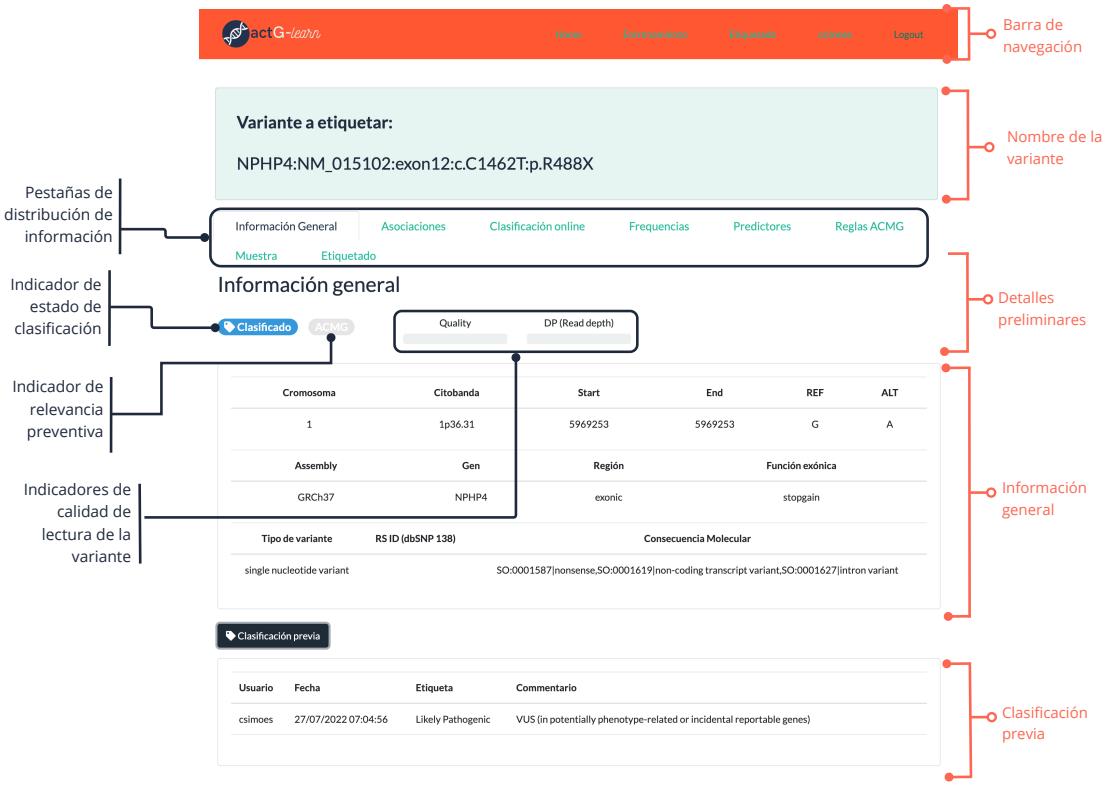


FIGURA 4.13: Vista de la información general en la interfaz. La información está organizada en siete pestañas que resumen la información de distintas fuentes externas (4.3) e internas hacia los usuarios. La parte superior de la pantalla (nombre de la variante y datos preliminares) se mantiene fija a medida que varían las pestañas de información. En la primera pestaña se despliega la información general, además de las clasificaciones previamente realizadas por los usuarios de la plataforma.

Información general.

En esta pestaña se ofrece información de alto nivel respecto a la variante, que representa los elementos más relevantes para su identificación, previo a acceder a detalles.
Entre los primeros aspectos que se observan son etiquetas, que ubican a la variante en determinadas categorías. Una de ellas es el etiquetado previo, si este elemento se encuentra resaltado en color, como se muestra en la Figura 4.13, entonces la variante ya fue evaluada previamente por algún usuario de la plataforma. En el caso de que la variante ya haya sido evaluada, los detalles de la etiqueta asignada se brindan al final de la vista, en donde se presenta a modo de tabla el usuario que evaluó la variante, la fecha en que fue realizada la última acción, la etiqueta asignada y el comentario adjunto.
Otra de las etiquetas en la vista general indica si el gen en el que se encuentra la variante es de relevancia preventiva. La indicación se basa en la versión 3.0 de la lista de genes indicada por la ACMG, a tener en cuenta como parte de la evaluación clínica de dalos de secuenciación de exoma y genoma, dada la accionabilidad médica de la condición asociada a los mismos [209].

Además de estos indicadores de características generales, se incluyen los datos que permiten identificar a la variante: ubicación (cromosoma, posición de comienzo, posición final, citobanda), cambio (alelo de referencia y alelo alternativo), genoma de referencia con el cual fue generada, gen en el que se encuentra, región de impacto en el genoma, función exónica o impacto de la variante, tipo de variante, ID obtenido de dbSNP, y la consecuencia molecular a través de ontologías [210].

Asociaciones.

En asociaciones se presenta información integrada a partir de OMIM [211], representada como un ejemplo en la Figura 4.14. El objetivo de esta pestaña es ilustrar la relación de genes y variantes con condiciones genéticas y sus principales características. Contar con una asociación de la variante o el gen con una afección y el modo de herencia implicado representan un dato clave al momento de considerar el posible efecto deletéreo de una variante [29].
En la pestaña se presenta un resumen de la información brindada por OMIM en forma de tabla, donde se encuentra, para cada fenotipo asociado al gen (representado por su numero MIM): descripción de fenotipo, modo de herencia, número MIM del fenotipo, clave de mapeo del fenotipo. Además, mediante links se puede acceder directamente al sitio web de OMIM para ampliar la información, tanto a la entrada del Gen como de cada fenotipo listado. Además, se ofrece un link para consultar el significado de cada número correspondiente a la clave del mapeo de fenotipo, en caso de que sea necesario.
El acceso a OMIM fue solicitado de forma gratuita para uso académico. Es importante destacar, que en la versión preliminar de la plataforma, donde el acceso es limitado y los usuarios pertenecen al grupo de trabajo asociado, es posible brindar esta información. En un escenario en el que el alcance de la plataforma sea mayor, se requerirá una licencia acorde a la distribución de la información.

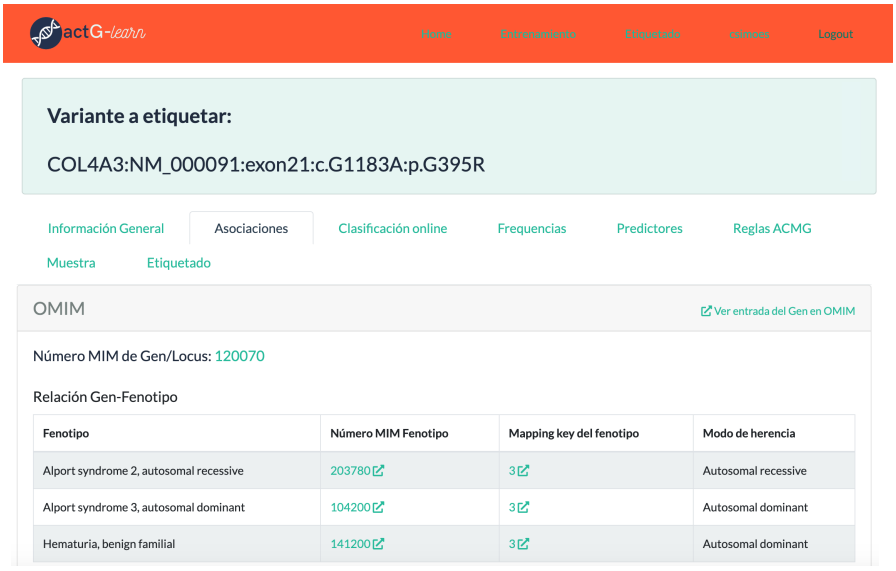


FIGURA 4.14: Vista de la información de OMIM para una variante. La información se presenta para cada gen, con su número MIM identificador correspondiente, a través del cual se puede acceder a la entrada en OMIM. Los datos se presentan en forma de tabla, donde las columnas identifican el fenotipo, número MIM del fenotipo, clave de mapeo del fenotipo y modo de herencia. Para cada número MIM de fenotipo se agrega un *link* externo a la entrada correspondiente en el sitio web de OMIM. Para las claves de mapeo se puede acceder a un *link* externo que permite acceder a la interpretación de cada clave en caso de que el usuario lo requiera.

Clasificación *online*.

En esta pestaña se ofrece información resumida de ClinVar relacionada la variante seleccionada a clasificar o etiquetar. El uso de las reglas de clasificación implica evaluar la evidencia reportada para la variante en cuestión, que permita comprender la relación entre el genotipo y el fenotipo clínico. Esto se puede obtener, en parte, del registro de las aserciones clínicas realizadas para cada variante identificada en el proceso diagnóstico, además de los datos fenotípicos asociados. ClinVar resume estas aserciones y la evidencia de forma estandarizada. Las observaciones clínicas realizadas además de tener una secuencia temporal que permiten identificar el avance en la interpretación, son asignadas con un estado de revisión que asocia las afirmaciones con determinados niveles de calidad a considerar en la posterior curación.
La vista en la pestaña se presenta en tres bloques: información general de ClinVar, tabla de interpretaciones, y tabla de conflictos. En la información general se resumen: nombre de la variante reportado en ClinVar, significado clínico consenso con el conteo de las interpretaciones individuales, estado de revisión, última evaluación realizada, número de usuarios que emitieron un ingreso, tipo de variante, identificador de alelo y código de variante de NCBI. La tabla de interpretaciones contiene un resumen de la evidencia aportada por cada acceso en ClinVar para la variante, y su representación es similar a cómo se muestra las interpretaciones y evidencias en la web de ClinVar. Las columnas están organizadas en: fenotipo reportado, significado clínico reportado, *status* de revisión, *submitter*, última fecha de evaluación, método de colección, origen (germinal o somático), y detalles de la evidencia. Para la columna de fenotipo reportado cada entrada tiene el link directo la entrada del fenotipo en MedGen [212]. La tabla de conflictos tiene como objetivo mostrar los pares de conflictos que han surgido entre etiquetas de los distintos *submitters*. En caso de que se requiera ahondar en la información de la variante en Clinvar, se cuenta con un link directo a la entrada de la variante en ClinVar [151]. Es importante considerar que esta sección puede cambiar a medida que la base de datos se actualice con actualizaciones en ClinVar.
La información brindada en esta pestaña le permite al usuario saber si fue evaluada por expertos y la concordancia de la interpretación. Si hay conflictos, el usuario puede ver la información de las interpretaciones que derivaron en conflicto.

Es importante considerar que lo que se presenta en esta pestaña puede variar entre los casos de uso, dada la información disponible para cada tipo de variante. Desde la toma de decisiones previas, en la instancia de aprendizaje el significado clínico final al que se llega en ClinVar no se muestra, ya que en algunos casos, puede sesgar el brindado de una evaluación final. Desde un punto de vista de disponibilidad de la información, como se mencionó en la Sección 4.3.1.2, las variantes conflictivas se destinan a la etapa de etiquetado únicamente, por lo que la presentación de conflictos no debería mostrarse en la instancia de aprendizaje.

Frecuencia poblacional.

En la pestaña de frecuencia poblacional se despliega información, siempre y cuando esté disponible, de distintas bases de datos de poblaciones (Tabla 4.3): gnomAD, 1000 Genomas, ExAC y ESP, incluyendo la frecuencia máxima observada en el conjunto. Lo primero que se observa es la frecuencia máxima, y luego se despliegan dos pestañas, cada una con una organización de las frecuencias observadas distinta. En una de las pestañas, las frecuencias se organizan por población: europea, africana, amerindia y asiática. En otra pestaña se organizan las frecuencias por proyecto: gnomAD, 1000 Genomas, ExAC y ESP. Las frecuencias se muestra e forma de gráficos de barra, con códigos de colores representado rangos de frecuencia, para que el usuario pueda identificar visualmente el aporte de la frecuencia en qué tan deletérea puede ser la variante (Tabla 4.4). El código de colores también se utiliza al momento de deplegar la frecuencia alélica máxima.

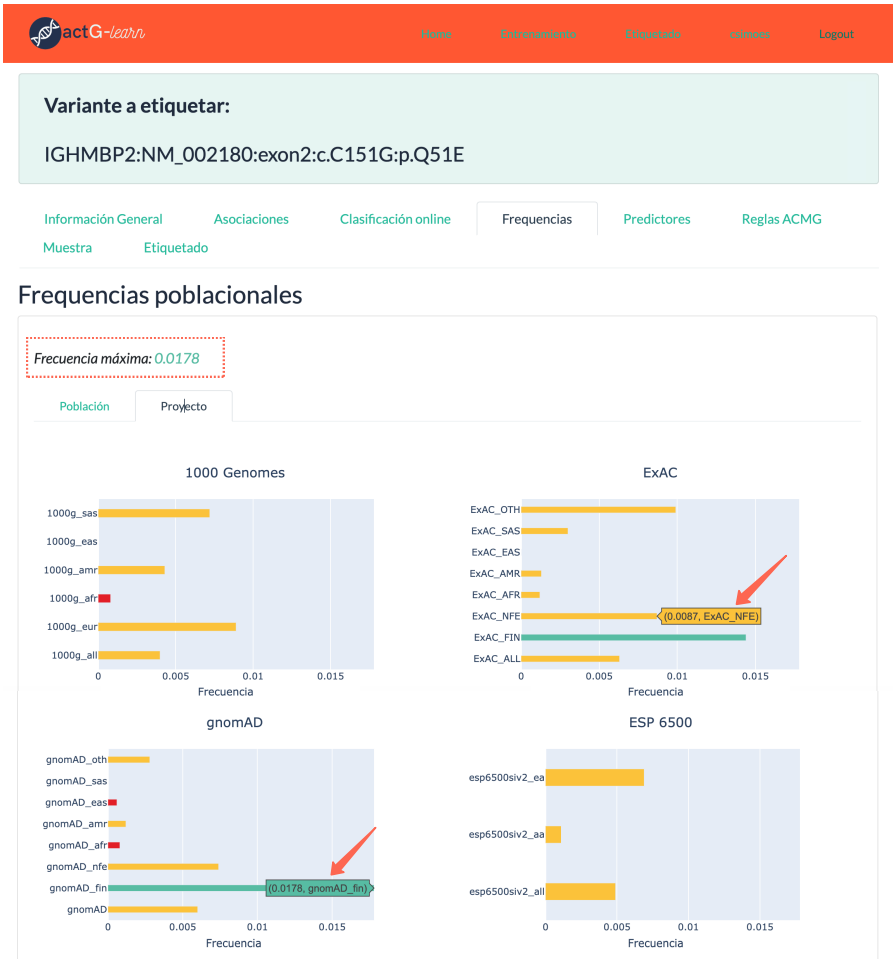


FIGURA 4.15: Ejemplo de vista de frecuencias alélicas en la plataforma. En la parte superior, remarcado en líneas punteadas se encuentra la frecuencia máxima. En el panel de ejemplo se muestran gráficamente las frecuencias agrupadas por proyecto. como puede observarse indicado con flechas, colocando el cursor sobre el gráfico se puede ver el valor de la frecuencia que se está observando. La escala de las gráficas se adapta a cada variante según la frecuencia máxima observada. El código de colores utilizado tanto para las gráficas de barra, como para la frecuencia máxima se presentan en la Tabla 4.4.

Rango de frecuencia (%)	**Color**
$f \leq 0{,}1$	
$0{,}1 < f \leq 1$	
$1 < f \leq 5$	
$f > 5$	

CUADRO 4.4: Código de colores para rangos de frecuencias alélicas. El código de colores se utiliza tanto para presentar la frecuencia máxima como para los gráficos de barras que despliegan frecuencias alélicas agrupadas por población y proyecto.

En lo que respecta a la aplicación de reglas ACMG, evaluar la frecuencia de una variante en una población de control o general es útil para valorar su patogenicidad potencial. Específicamnete, al usuario le servirá para evaluar si se cumplen los criterios BA1 (frecuencia alélica superior al 5 %), BS1 (frecuencia alélica es superior a la esperada para el desorden) y PM2 (la variante se encuentra

ausente en los controles).

Predictores.

En la pestaña de predictores se ofrecen los valores correspondientes a herramientas de predicción *in silico*, destinadas a la asistencia en la interpretación de variantes. Hay dos sub-pestañas dentro de Predictores que agrupa a las 13 herramientas (Tabla 4.3) en predictores funcionales o de impacto físico-químico (7), de conservación evolutiva (4), de efecto en el splicing de ARN (1).
La información se presenta en forma de tabla, donde para cada predictor se indica el *score* y la predicción asociada. Para cada predictor se brinda la predicción transformada desde su código, para reducir el trabajo al usuario al momento de utilizar la información (Figura 4.16).
La evaluación de la patogenicidad usando reglas, es asistida por estos predictores en los criterios PP3 (múltiples líneas de evidencia computacional apoyan un efecto deletéreo sobre el gen o su producto), BP4 (múltiples líneas de evidencia computacional sugieren que no hay impacto sobre el gen o su producto), PP2 (variante *missense* en un gen con una tasa baja de variaciones *missense* benignas y en el que las variantes *missense* son un mecanismo común de enfermedad) y BP1 (variante *missense* en un gen del que se sabe que causa principalmente variantes truncantes de enfermedad).

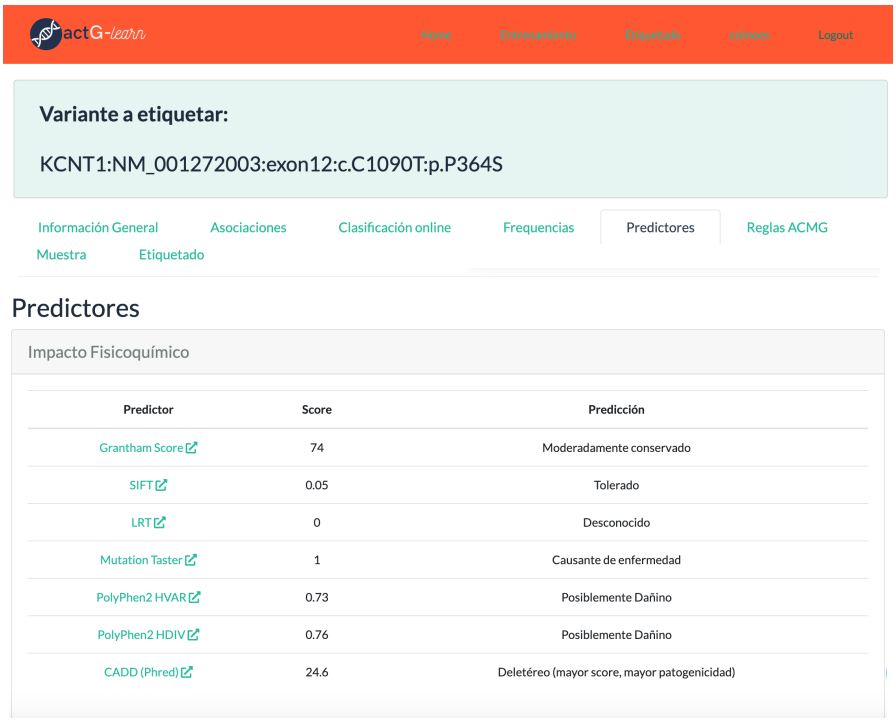


FIGURA 4.16: Ejemplo de vista de predictores *in silico* en la plataforma. Se presentan solamente los predictores de impacto fisicoquímico. Para cada predictor se presenta el *score* y su predicción correspondiente. En la mayoría de los casos la predicción de traduce del código generado por la anotación, por lo que en la mayor parte de los casos no se realizó un cálculo de la predicción final. Muchos predictores no cuentan con un umbral definido para establecer qué tan deletérea es la variante, por lo que en esos casos se hace una estimación aproximada de acuerdo al estado del arte. En el caso de los predictores en conservación evolutiva, no se genera una predicción que de a entender el impacto de la variante, sino qué tan conservada es la posición que se vio modificada.

Información de la muestra.

En caso de que la evaluación de la variante no sea en abstracto y se encuentre en el contexto de una muestra o paciente, se puede contar con datos clínicos del individuo en el que aparece la variante. En este caso, se presenta este tipo de información en una pestaña asociada a la muestra. En este caso se brinda la información de: el fenotipo observado, el modo de herencia, la cigosidad para la variante observada, el sexo del sujeto, si hay familiares afectados, la etnicidad del sujeto, y el origen de la muestra (de dónde fue extraída). En esta instancia la evaluación del modo de herencia y la cigosidad del paciente resulta fundamental. Esta etapa de análisis implica, por ejemplo, evaluar si la cigosidad en la que se encuentra la variante tiene sentido respecto al modo de herencia que presenta el fenotipo observado. Esta instancia generalmente requiere un análisis manual.

Reglas.

Una vez evaluada toda la evidencia que se recolectó para la variante analizada, se puede presentar un resumen de una clasificación automática según las reglas ACMG/AMP. La visualización de las reglas de la interfaz depende del caso de uso en el que se encuentre el usuario.
En caso de encontrarse etiquetando variantes con conflicto o tipo desconocido, se despliegan las reglas, señalando el color aquellas que se cumplen (Figura 4.17). En caso de que no se tenga una predicción de reglas para la variante, las reglas se presentan en color gris. La información de las reglas que cumple cada variante se obtiene de Intervar [86] a través de la anotación con ANNOVAR, y se aplica dependiendo de la información disponible para cada variante. Además, se pone a disposición al usuario información respecto al uso de las guías a los efectos de comprender el origen de la clasificación y permitirle evaluar si la clasificación brindada es la correcta, o quiere modificar su postura.
En caso de encontrarse clasificando las variantes con un objetivo de aprendizaje, solo se despliega la información respecto al uso de variantes con el objetivo del que el sujeto logre clarificar solo la variante. Luego que el usuario ingresa su veredicto, parte del *feedback* incluye la pertinencia o no de la etiqueta dependiendo de las reglas que cumple la variante y el consenso final al que se debería llegar.
En la Figura 4.17 se presenta un ejemplo de vista parcial de las reglas en la plataforma.

FIGURA 4.17: Ejemplo de vista de reglas ACMG en la plataforma. La vista se encuentra interrumpida en secciones (líneas punteadas) a modo de reducir la vista general con la descripción de cada regla. En la parte superior de la vista se presenta un resumen de las reglas que se cumplen. En la parte inferior se muestra la lista de todas las reglas propuestas por ACMG: en colores, las reglas que se cumplen, en blanco las que no se cumplen. A través de *links* se puede acceder al artículo con el detalle de las reglas ACMG, o a refinamientos de las mismas [44] [78]

Reporte de la clasificación.

La última pestaña corresponde a la del reporte de la clasificación. En esta instancia el usuario elige entre las etiquetas: Benigna, Probablemente Benigna, Patogénica, Probablemente Patogénica, Variante de significado incierto, Factor de Riesgo y Respuesta a Drogas.
En caso de que el usuario se encuentre en la etapa de etiquetado, la etiqueta se acompaña de comentarios asociados a las mismas, a modo de explicación de algunos sucesos. Estos comentarios ya se encuentran pre-definidos, a los efectos de mantener control sobre lo que ingresa el usuario:

- *Considerado benigno o sin importancia clínica.* Este comentario incluye las etiquetas benigna y probablemente benigna, pero también VUS en heterocigotos para recesivas.
- *Se considera benigna ser es intrónica.* Se usa cuando la variante tiene una frecuencia baja pero no hay otros datos que indiquen posibilidad de patogenicidad.
- *VUS (en genes potencialmente relacionados con el fenotipo o de notificación incidental).*
- *No corresponde al fenotipo estudiado* (si se cuenta con información de la muestra estudiada).

- *No se conoce ningún fenotipo o riesgo asociado al gen.*
- *Heterocigoto en AR o XR (XX).* Este comentario se utiliza en caso de etiquetar como probablemente patógénico, patogénico o VUS.
- *Incidental*, en caso de que se indique patogénico o no patogénico a una variante no relacionada con el fenotipo estudiado.
- *Candidata.* En caso de que se considere probablemente patogénica o patogénica a una variante relacionada con el fenotipo estudiado.

Los comentarios pre-definidos fueron acordados en conjunto de los genetistas asesores en la plataforma. Su generación se considera de utilidad a aplicaciones a futuro, que permitan identificar patrones de desempeño en la clasificación asociados, aunque en la implementación actual, no son utilizados a posteriori, más que como asistencia.
Luego de emitida la etiqueta y comentario, se ingresa el reporte y se cierra el proceso.
En caso de que el usuario esté en la instancia de aprendizaje, solamente emite la etiqueta, y luego de realiza el reporte el sistema le brinda un *feedback* respecto al desempeño, y un resumen de las reglas que se cumplen para la variante.
Además, se cuenta con un botón que permite descargar la variante junto a su anotación en formato .xls.

4.3.2.3. Desarrollo de la interfaz en Dash

Como fue mencionado anteriormente, la interfaz fue implementada utilizando Dash, herramienta que le da la estructura a todo el proyecto. En este trabajo en particular, se usó la versión 1.8 de Dash, sobre Python 3.7.3. La aplicación web consiste en una aplicación Dash multi-página con*login*, la cual utiliza los servicios web del servidor.

La aplicación web está basada en el proyecto `dash-auth-flow` de Russell Romney [213]. Este proyecto brinda a la aplicación el flujo de autenticación de usuario incluido en Dash, el cual dispone de páginas y funciones para poder realizar el flujo de autenticación completo, incluyendo: inicio, inicio de sesión, cierre de sesión, registro, contraseña olvidada y cambio de contraseña. Esta aplicación, a su vez usa *flask-login* en el *back-end*, basándose en el código de `dash-flask-login` [214], que permite la autenticación de usuarios a través de una base de datos SQLite3 [215].

En general las aplicaciones Dash constan de dos grandes partes (sin contar la importación de paquetes, creación de la aplicación y ponerla a correr), y la primera es el diseño o *layout*, que se centra en el aspecto de la aplicación [216] [181]. Una vez tomada la estructura inicial mencionada en el párrafo anterior, se procedió a definir dicho diseño, el cual se encuentra principalmente contenido en las páginas en las que se organiza el proyecto. Sobre estas páginas se creó el *layout* presentado de forma general en la Figura 4.10, para el cual se utilizaron como base los componentes HTML de *Dash Core Components* (DCC), *Dash HTML Components*, y DBC utilizando Bootstrap 4. *Dash HTML Components* proporciona wrappers en Python para elementos HTML, y en este caso, la librería fue utilizada principalmente para crear *containers* genéricos (`html.Div`), párrafos (`html.P`), encabezados (`html.H1 a H6`), enlaces (`html.A`), saltos de línea (`html.Br`), elementos de tablas (`html.Tr`,`html.Td`), agregar imágenes (`html.Img`), entre otros. DCC por otra parte permite generar componentes de nivel superior como controles y gráficos, y en el proyecto se utilizó específicamente para la generación de elementos interactivos tales como desplegables (`dcc.Dropdown`), gráficas (`dcc.Graph`) (por ejemplo, las gráficas de barra de frecuencias) y para acceder a distintos *paths* dentro de la aplicación (`dcc.Location`). Gran parte del desarrollo de la interfaz también se basó en componentes de Bootstrap, un *framework front-end*, que permite brindarle un diseño uniforme, visualmente más atractivo (relativamente) y "profesional" a la interfaz. Además, las características de diseño "responsivo" de

Bootstrap permiten que la aplicación se muestre de forma correcta y coherente en dispositivos variados con distintos tamaños de pantalla. En este proyecto en particular, DBC se utilizó, por ejemplo, para obtener la *stylesheet* de la plataforma, para estructurar el *layout* (`dcc.Container`, `dcc.Col`, `dcc.Row`), construir barras de navegación (`dbc.Navbar`, `dbc.NavItem`, `dbc.NavLink`), dar *feedback* dependiendo del contexto a los usuarios (`dbc.Alert`), agregar botones (`dbc.Button`), componentes de diálogo (`dbc.Modal`), pestañas autocontenidas (`dbc.Tab`), "tarjetas´´ de contenido (`dbc.Card`), crear tablas (`dbc.Card`), componentes de formulario para controlar entradas, tales como etiquetas, controles, texto de ayuda opcional y mensajes de validación en caso de formularios (`dbc.FormGroup`, `dbc.Label`, `dbc.Input`, `dbc.FormText`); entre otros componentes.

Mientras que la primera parte de la aplicación consiste en el diseño de la interfaz, la segunda parte describe la interactividad de la misma. Luego de generado el *layout*, se implementó la interacción del usuario con la interfaz, la cual en Dash se realiza a través del uso de *callbacks*. Los *callbacks* son funciones que permiten enlazar los elementos de la interfaz entre si, brindando la funcionalidad deseada. Son llamadas para afectar la apariencia de un elemento HTML (la salida) cada vez que el valor de otro elemento (la entrada) cambia. El funcionamiento de los *callbacks* es dinámico, y no depende de que la página vuelva a cargarse [181] [217]. La plataforma cuenta con cerca de 30 *callbacks* que definen su interactividad, los cuales pueden llegar a depender de varias entradas, y cuya mayoría devuelve muchas salidas.

4.3.3. Backend

4.3.3.1. Base de Datos

4.3.3.1.1. Obtención y pre-procesamiento de datos externos

Los datos que se toman como insumo basal para la plataforma provienen principalmente de ClinVar [218]. Para la base de datos se descargaron los siguientes conjuntos de datos de dicho recurso:

- Variantes (`clinvar_X.vcf.gz`): un archivo en formato VCF (versión 4.1) que contiene el conjunto de datos de ClinVar, en el que cada variante se representa como una posición mapeada contra el genoma de referencia GRCh37.
- Resumen de variantes (`variant_summary.txt`) un archivo delimitado por tabulaciones basado en cada variante en una ubicación del genoma para la que se han enviado datos a ClinVar. Los datos de la variante se reportan para cada ensamblaje, por lo que la mayoría de las variantes tienen una línea para GRCh37 y otra línea para GRCh38.
- Resumen de acceso/reporte (`submission_summary.txt`): un archivo delimitado por tabulaciones que contiene un resumen de la interpretación, fenotipos, observaciones y métodos comunicados en cada reporte a ClinVar.
- Resumen de interpretaciones conflictivas (`summary_of_conflicting_interpretations.txt`): un archivo delimitado por tabulaciones que reporta todos los pares de reportes que generan conflictos en sus etiquetas.
- Resumen de organizaciones (`organization_summary.txt`): un archivo delimitado por tabulaciones que contiene un resumen de las organizaciones que emiten reportes de variantes en ClinVar.

Además, se obtienen los siguientes conjuntos de datos de OMIM:

- `mim2gene.txt`: un archivo delimitado por tabulaciones que permite vincular números MIM con IDs de genes (NCBI, Ensembl, HGNC).

- `genemap2.txt`: un archivo delimitado por tabulaciones que contiene una sinopsis de OMIM del Mapa Genético Humano, incluyendo información adicional como coordenadas genómicas y herencia (de gran importancia).
- `morbidmap.txt`: un archivo similar al anterior, ordenado alfabéticamente por trastorno.

Antes de ser ingresados a la base de datos, los datos de variantes se acondicionan como de describe a continuación:

1. Las variantes en formato VCF pasan por un primer proceso de anotación con ANNOVAR, descrito anteriormente, mediante el cual se obtiene como salida un archivo delimitado por tabulaciones. Además, se utiliza ANNOVAR para realizar el cálculo del Grantham Score, el cual se realiza separado de la anotación.
2. Los archivos accesorios que lo requieran, son adaptados a contener las líneas que corresponden únicamente a la referencia utilizada.
3. Se comienza a generar la tabla de variantes, uniendo el archivo anotado con los archivos accesorios adquiridos:
   - OMIM: `mim2gene.txt` (para obtener número mim, tipo, gen, e identificadores en otros recursos), `morbidmap.txt` (para obtener principalmente los fenotipos), y `genemap2.txt` (para ampliar información complementaria a los fenotipos).
   - Tabla que contiene el Grantham Score calculado.
   - ClinVar: `genemap2.txt` (para ampliar la información respecto al reporte de cada variante).
4. Se calcula la frecuencia máxima a partir de la anotación de frecuencias alélicas.
5. Se agregan columnas con datos que inicialmente no están incluidos en ClinVar pero interesa conseguir para otros tipos de variantes de muestras identificadas (cigosidad, columnas de calidad, número de variantes en la que la muestra es homocigota/heterocigota para la referencia/alelo alternativo, etc.)
6. Se agrega un identificador interno para las variantes, el cual será la clave en la tabla una vez que esté en la base de datos.
7. Se agregan etiquetas para separar variantes conflictivas de no conflictivas.
8. Se acondiciona la tabla para que la estructura sea la esperada en la base de datos.

Además de la tabla de variantes, los otros archivos adquiridos de ClinVar son acondicionados a los efectos de que su estructura sea acorde a las tablas que se quieren lograr: se acondicionan los nombres de las columnas, se agregan los identificadores de variantes correspondientes, se controla la existencia de símbolos no deseados que puedan dificultar la importación de datos, etc. Finalmente las tablas finales generadas son importadas en MySQL.
Cabe destacar que los datos de ClinVar son actualizados semanalmente, cada lunes. Si bien se cuenta con una estrategia de automatización del acceso a datos y su procesamiento, actualmente el proceso de curado se hace de forma manual, y la actualización se hace de forma mensual.
Si bien el estado basal de la plataforma es que su funcionamiento inicie únicamente con variantes pertenecientes a ClinVar, se utilizan otros recursos con variantes también, que en un principio son consideradas como "no conflictivas", destinadas a la etapa de aprendizaje. En este sentido, se cuenta con un conjunto de variantes obtenidas del proyecto 1000 Genomas, el cual tuvo el mismo procedimiento descrito anteriormente.

4.3.3.1.2. Estructura

La base de datos fue finalmente implementada usando MySQL 8.0.22. Los elementos y sus relaciones principales de la base de datos se muestran en la Figura 4.18. En total la base de datos consiste en 14 tablas, cuyo detalle de atributos y relaciones se especifican en Apéndice C. El modelo de datos se centra en la clasificación de variantes, y en el entrenamiento de usuarios. Este modelo de clasificación está basado en datos y contexto de la variante que contribuyen a indicar su nivel de patogenicidad.

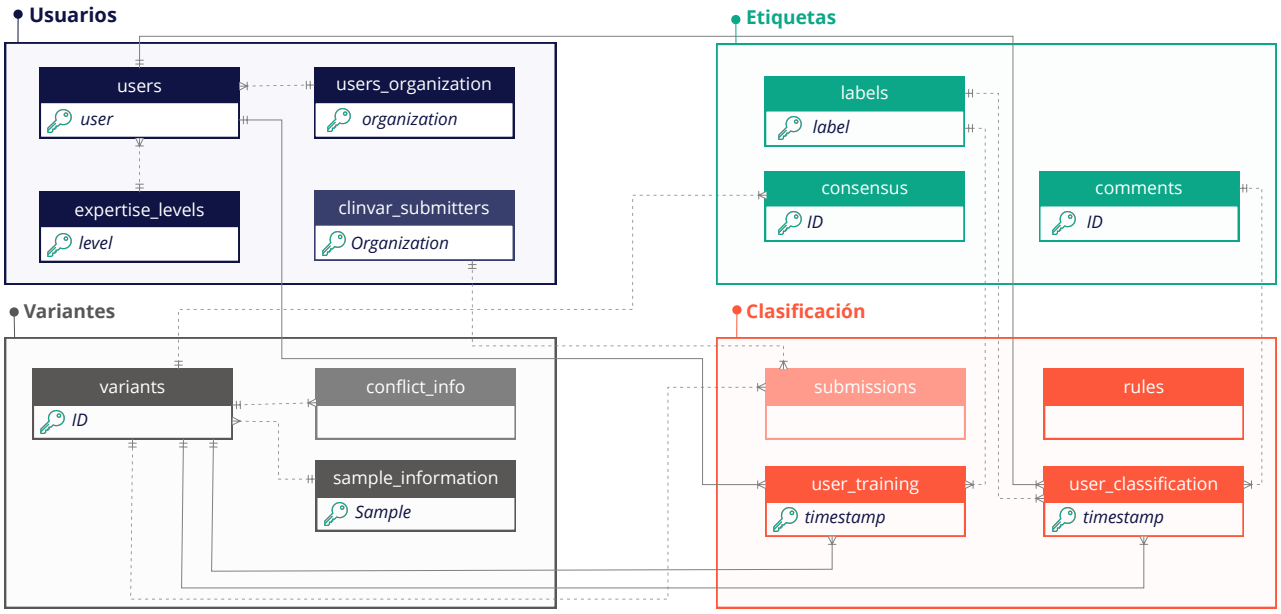


FIGURA 4.18: Diagrama de la base de datos implementada en la plataforma. Las 14 tablas que contiene la base de datos se describen según su organización en vistas a cuatro niveles: usuarios, etiquetas, variantes y clasificación.

- Usuarios: a nivel de usuarios hay cuatro tablas destinadas a su descripción. A grandes rasgos, estas tablas pueden dividirse en aquellas que guardan información exclusiva de la plataforma, y aquellas que guardan información original extraída de ClinVar. Respecto a la información propia de la plataforma, se creó una tabla que almacena la información de los usuarios registrados en la misma, los cuales participan activamente en los procesos de etiquetado y clasificación. Entre la información que almacena esta tabla, se encuentran aquellos datos que participarán directamente en los modelos de aprendizaje, consenso de etiquetas y acceso a datos, tales como el nivel del usuario, nivel de acceso e información de actividad en la plataforma. Esta tabla está relacionada a dos tablas: la que contiene los cinco niveles de usuario detallados anteriormente, y la que guarda información de la organización a la que pertenece el usuario. Esta última tabla se ha creado a los efectos de contar con un nivel de *status* más general al momento de evaluar el estado de análisis de una variante, además de brindar un soporte al nivel de pericia del usuario. En esta tabla se almacena, entre otras cosas, la cantidad de clasificaciones y etiquetas aportadas, la actividad temporal, y el nivel más alto de usuario. Como fue comentado anteriormente, la tabla de usuarios y organizaciones se mantienen separadas de la tabla de *submitters* de ClinVar. Esta última almacena información de las organizaciones que emiten una clasificación en ClinVar. En un principio, estas tablas se colocan por separado, dado que contienen distintos tipos de información, respecto a aquella con la que se cuenta en la plataforma (por ejemplo,

los mecanismos de colección de información), además de contener datos de organizaciones que en su gran mayoría no son activas en la plataforma actualmente. En un escenario en el que se trabaja con una base de datos relacional, contener una tabla de organizaciones o usuarios única implicaría que muchos campos queden vacíos en gran parte de los datos. A futuro, atendiendo a posibles modificaciones en la estrategia de distintos aspectos del *backend*, se podría contemplar unificarlo, atendiendo además a la posibilidad de que pueda ampliarse internamente a más grupos. En un principio, la información de esta tabla se considera importante al momento de que el usuario de la plataforma considere las evaluaciones previas externas realizadas por la variante.

- Variantes: la base de datos contiene una gran tabla que almacena toda la información referida a cada variante, que incluye los datos poblacionales, experimentales, computacionales y evidencia basada en literatura, explicados anteriormente. En un principio, las variantes obtenidas de distintas bases de datos y mecanismos (incluidas por el administrador o el usuario). Dentro de esta tabla se identifican dos grandes tipos de variantes seleccionadas según la funcionalidad de la plataforma a la que servirán. Atendiendo a que el objetivo del etiquetado es desambiguar variantes con etiquetas previas conflictivas, el tipo de las variantes se toma para separarlas en dos grupos y así destinarlas. Aquellas que se consideran como conflictivas, están a su vez vinculada a una tabla que contiene el detalle de los conflictos generados. Esta última tabla es obtenida directamente de ClinVar y contiene los pares de interpretaciones que dieron lugar a conflicto, para cada variante. Se considera que contar con las evaluaciones realizadas previamente puede ser útil para la evaluación actual. Además, las variantes pueden estar asociadas a una muestra o un paciente determinado. En este caso, toda la información de la muestra podría categorizarse por separado, sin embargo a ser un enfoque inicial, se la mantiene vinculada a las variantes. En un contexto donde las variantes provienen de bases de datos públicas que no aportan información de la muestra, la mayoría de las variantes cargadas en la base de datos actualmente no cuentan con esta información. En casos donde se cuente con información de la muestra en la cual se identificó la variante, la misma se almacena en otra tabla.

- Clasificación: La clasificación de los usuarios, en cada instancia funcional de la plataforma, está contenida principalmente en dos tablas. Una de ellas está destinada al almacenamiento de las clasificaciones emitidas por usuario sobre cada variante en la instancia de aprendizaje. En este caso es de interés almacenar, además de la etiqueta y los datos del usuario, un registro de si la etiqueta asignada es correcta o no. Esto permite actualizar el nivel del usuario, en función de su desempeño a lo largo del tiempo. La otra tabla que almacena una clasificación de usuario participa en la instancia de etiquetado. En este caso también se almacena la etiqueta, datos de usuarios y temporales, agregando comentarios asociados a la etiqueta, y quitando la evaluación de la etiqueta correcta. Este último aspecto se debe a que en realidad, no se conoce con certeza la etiqueta correcta, y solo se almacena la información brindada por el usuario. Las etiquetas y los comentarios brindados por los usuarios ya se encuentran pre-definidos y dentro de la base de datos, con el fin de evitar distintos criterios y posibles errores en el ingreso manual que dificulten el posterior análisis. En cada instancia del proceso se requiere la información detallada de las reglas ACMG, tanto para brindar *feedback* al usuario, como para guiarlo en el proceso de clasificación. Para ello, se cuenta con una tabla que cuenta con el detalle de cada regla para su presentación. Si bien se almacena cada etiqueta emitida por cada usuario, una misma variante puede recibir distintas etiquetas, tanto de distintos usuarios como del mismo en distintas instancias de tiempo.

- Etiquetas: Las etiquetas que emiten los usuarios, almacenadas en las tablas explicadas en el ítem anterior, se extraen de una tabla que las define. Estas etiquetas son las mencionadas anteriormente, y se pueden diferenciar según el proceso que se encuentre realizando el usuario (aprendizaje o etiquetado). Teniendo esto en cuenta, en la etapa de etiquetado se permiten

etiquetas más variadas, atendiendo a la incertidumbre sobre el tipo real de las variantes. Se cuenta además con una tabla que almacena el consenso de etiquetas: una variante etiquetada por distintos usuarios, con distintos niveles y en distintas instancias. La etiqueta consenso es calculada a partir de: último nivel de usuario, tiempo de actividad, avance, soporte de la variante original (de ClinVar).

4.3.3.2. API

La API en la aplicación es la responsable de las interacciones con la base de datos descrita a modo general en la sección 4.3.3.1, además de los modelos predictivos. La API fue implementada utilizando `Flask 1.1.2`. Para la conexión con la base de datos desde Python se utilizó `mysqlclient 2.0.1`, con el módulo `MySQLdb` como interfaz [219]. Las consultas MySQL realizadas a la base de datos se realizan devuelven su respuesta a la API a través de archivos JSON, mediante el paquete `simplejson 3.17.2`, que funciona como un codificador y decodificador JSON dentro de Python [220]. La documentación de la API fue generada utilizando Swagger (`flask-swagger-ui 3.36.0`) [221] [222].

La API implementada se vale principalmente de los dos métodos HTTP más comunes utilizados para enviar información: GET (para recuperar datos), y POST (para actualizar o almacenar datos) [223]. Las rutas de la API se pueden agrupar según sus objetivos en:

- Solicitud de variantes: consiste en rutas de la API utilizadas para solicitar y devolver conjuntos de variantes. A través de estas rutas se pueden devolver, por ejemplo, variantes específicas a través del identificador interno de la plataforma, conjuntos de variantes de tipos específicos (conflictivas, no conflictivas), y se puede determinar el modo de selección de variantes: aleatorio o no aleatorio.
- Solicitud de usuarios: son las rutas a través de las cuales se solicita y devuelven acciones de los usuarios, o información de los mismos. A través de estas rutas es que se obtienen estadísticas asociadas al desempeño de los usuarios (a través de su identificador único), se envía una acción de etiquetado o clasificación en modo de aprendizaje, y se ingresa al usuario a la plataforma a través de sus datos.
- Solicitud de etiquetas: consiste en las rutas destinadas a la solicitud y devolución de etiquetas disponibles y/o realizadas a las variantes, dependiendo de su tipo.
- Solicitud de comentarios: consiste en las rutas destinadas a la solicitud y devolución de comentarios disponibles y/o realizadas a las variantes, que van asociados a una etiqueta determinada y complementan la clasificación.
- Solicitud de niveles: consiste en las rutas destinadas a poner a disposición los niveles de los usuarios que se desempeñan en la plataforma.
- Solicitud de muestras: consiste en las rutas destinadas a poner a disposición la información de las muestras, o de ingresar la misma.

4.3.3.3. Lógica de transformación de datos

Hay dos aspectos que se consideran de importancia dentro de la funcionalidad de la plataforma, ya que modifican el comportamiento de la misma, de los datos que son manipulados y de los usuarios que participan: cómo se aborda el aprendizaje del usuario y cómo se agregan las etiquetas brindadas. En ambos casos, el abordaje que se siguió es igual que el de la plataforma: inicial y con una funcionalidad mínima. En este sentido, el objetivo consistió en generar una base funcional, para que una vez que se comience a recolectar una cantidad de datos considerable, estos enfoques comiencen a evaluarse y mejorarse. Cabe aclarar es que a lo largo del documento se ha hablado indistintamente de “expertos”

tanto para referirse al contexto de conjunto de personas o usuarios que participan en el etiquetado o clasificación, como al máximo nivel de dichos usuarios en términos de pericia. En la sección actual se hará referencia al término únicamente en le contexto de nivel de pericia, a los efectos de no generar confusiones.

4.3.3.3.1. Instancia de aprendizaje

El aprendizaje de los usuarios de la plataforma se aborda desde un enfoque preliminar, que abarca 3 aspectos: la información presentada al usuario, el manejo de los datos presentados al usuario, y la actualización de los niveles de pericia.

Datos presentados al usuario

El primer aspecto a evaluar es la información presentada. Al momento de describir la interfaz ya se introdujo el modo en el que se presenta la información, con el fin de que el usuario pueda, a través del orden y la categorización de la información, buscar la forma de aplicar las reglas de clasificación estandarizadas. Además de cómo se muestra la información, y qué partes de la misma se dejan disponibles, los datos son asignados dependiendo de la complejidad de los mismos. Esto quiere decir que los conjuntos de variantes están separados por una categorización preliminar de su complejidad, y dependiendo del nivel de usuario se asignan distintos conjuntos, a los efectos de que a medida que el usuario aumente de nivel, aumente la complejidad de las variantes que se le presentan.
Las categorías establecidas para los tipos de datos se encuentran resumidas en la Tabla 4.5). En el caso del usuario en un nivel de iniciación (novato), se asignan variantes que hayan sido clasificada anteriormente como Benignas o Patogénicas por un panel de expertos en ClinVar, o que la evidencia para dicha etiqueta sea fuerte en caso de que provengan de otras bases de datos. En este último caso, se considera, por ejemplo, si provienen de bases de datos específicas referidas a algunas enfermedades en las que la variante se haya identificada como causal o factor de riesgo. Dado que el usuario está dando sus primeros pasos en la tarea, la misma debe reducir las complejidades de la incertidumbre intrínseca de la tarea, de modo que el foco se encuentre en las características relevantes que lleven a un significado clínico. En este sentido, se considera que un abordaje preliminar puede ser seleccionar las variantes que se encuentren en los "extremos" del rango, que por lo general cuentan con características y evidencia fuertes para sustentar la clasificación. Se trata de brindar las variantes que tengan la mayor cantidad de información posible en todos los aspectos abordados por las reglas. Si bien (en un contexto de aplicación de reglas) la clasificación de una variante como Patogénica confiere una mayor complejidad, es importante que esto se entrene desde el inicio. En el caso de tratarse de un usuario principiante, si bien ya cuenta con una experiencia mínima, la cual la permitió acceder a un nuevo nivel, aún se encuentra enfrentado a sus primeras clasificaciones. En este caso se comienzan a agregar variantes que tengan un nivel de soporte menor para la etiqueta en cuestiones de evidencia, de esta forma se incorporan variantes Probablemente Benignas y Probablemente Patogénicas. Atendiendo a que un usuario competente ya sea capaz de identificar casos en los que no es posible determinar la patogenicidad de una variante se incorporan las variantes VUS en este nivel. Como se explicó en instancias anteriores, los usuarios avanzados y expertos, son en general requeridos en el etiquetado. Si embargo esto no los exime de la instancia de entrenamiento, principalmente a quienes aún no han alcanzado el nivel más alto. En este sentido, se agregan variantes con tipos que no se incluyen en las reglas ACMG, pero que se espera que un usuario con experiencia, y teniendo en cuenta la evidencia de las mismas, pueda reconocer.

	Novato	Principiante	Competente	Avanzado	Experto
Benigna	✔	✔	✔	✔	✔
Probablemente Benigna		✔	✔	✔	✔
VUS			✔	✔	✔
Factor de riesgo				✔	✔
Respuesta a drogas					✔
Probablemente patogénica		✔	✔	✔	✔
Patogénica	✔	✔	✔	✔	✔

CUADRO 4.5: Resumen de los tipos de variantes asignados a cada nivel de usuario en la instancia de aprendizaje. En columnas se encuentran los niveles de usuario generados, y en las filas los 7 tipos de variantes seleccionados para trabajar en la etapa de entrenamiento de usuarios. Si bien no se cuenta con un modelo de complejidad de variantes que sustente la decisión, se considera que la asignación contemple una idea preliminar de complejidad de variantes, la cual debe aumentar a medida que aumente el nivel de usuario. Esta complejidad se asocia a qué tanta información se cuente para las variantes, y qué tanto se puedan aplicar las reglas ACMG. En este sentido las variantes con una patogenicidad más marcada, como Benignas o Probablemente Benignas, son asignadas a usuarios novatos, y a principiantes se les agregan las Probablemente Patogénicas y Probablemente Benignas. Cuando el usuario ha alcanzado un nivel de competencia considerable, entonces se agregan variantes de significado incierto. Los niveles más elevados a su vez pueden recibir variantes con tipos que no se encuentran incluidos en los estándares de ACMG, como factores de riesgo y variantes que modifiquen la respuesta a drogas [44]. En esta instancia no se asignan variantes con interpretación conflictiva, las cuales están únicamente destinadas a la instancia de etiquetado.

Como fue mencionado anteriormente, dado que son variantes que se destinan a la instancia de entrenamiento, cuentan con niveles de revisión altos.
La forma en que los usuarios reciben los datos tiene un abordaje inicial en cuanto a mecanismos que aseguren el aprendizaje. En la instancia actual de la plataforma dicho abordaje implica incluir en la plataforma la técnica de repetición espaciada, mejor conocido como *spaced repetition* [224] [225]. El objetivo de contar con esta técnica, es que, si bien la idea no es que el usuario memorice las variantes y sus características, si logre retener las condiciones que llevan a emitir una determinada etiqueta a través de la exposición repetida a la información. Para abordar este aspecto, se hace una selección de variantes por cada usuario, centrándose en aquellas que no logra identificar el significado clínico, además de las que no han sido evaluadas por el mismo usuario en un tiempo determinado. Luego se aplica un modelo que es capaz de predecir cuándo se ha olvidado algo visto previamente, tendiendo en cuenta que no se ha visto con mucha frecuencia, o muy recientemente. La frecuencia de repetición y los intervalos son determinados mediante el paquete de Python`SuperMemo2` (de "Super Memory"), que implementa el algoritmo de *spaced repetition* SM-2, que permite calcular la próxima fecha de repaso de la tarea que se esté aprendiendo [226] [227]. Dicha fecha se actualiza para cada variante en la base de datos.

Actualización de niveles de usuario

En la implementación actual de la plataforma, los niveles se seleccionan por el administrador, mediante la asistencia de un experto que ayuda a determinar los niveles iniciales en la plataforma. Este funcionamiento se adapta a la etapa de modo de prueba en un grupo reducido. A partir de allí, los niveles pueden ir cambiando dependiendo del desempeño de los usuarios. En la versión actual hay dos aspectos que se consideran al momento de evaluar un usuario, y determinar su avance o retroceso de nivel: actividad (frecuencia con la que participa en la plataforma en cualquiera de sus casos de uso) y desempeño (qué tan bien avanza en la instancia de aprendizaje). Si bien se podría contemplar un primer abordaje que incluya también la fluidez en la tarea (qué tanto demora emitiendo una etiqueta), es aún más difícil que los otros aspectos mencionados tener una estimación basal sobre la

cual comparar.
Los niveles están asociados de alguna manera a habilidades, y las mismas deben ser superadas antes de pasar a un nuevo nivel. Además, atendiendo a que falta de práctica puede tener como consecuencia el olvido o pérdida de estas habilidades, los niveles deben estar asociados al tiempo que las actividades dejan de hacerse. En la implementación actual, se utiliza un sistema de puntos interno para actualizar los niveles, el cual está basado en "puntos de experiencia", a menudo abreviados como XP[8]. Los puntos de experiencia, permiten cuantificar la progresión del usuario, y son muy utilizados en plataformas de enseñanza a través de estrategias de gamificación [228, 229]. Los niveles de usuario se asocian a rangos de XP determinados, los cuales inicialmente se tomaron siguiendo la experiencia de otras plataformas de gamificación [230]. En la Tabla 4.6 se presenta un resumen de los rangos de XP para cada nivel, donde se puede observar que la progresión es no lineal. A los efectos de generar que el usuario se motive e involucre en la tarea, se necesitan obtener relativamente pocos puntos para moverse en los niveles iniciales, sin embargo los niveles superiores requieren de más puntos para ser alcanzados. A cada usuario que se carga en la plataforma con un nivel inicial, le es asignada la cantidad de XP correspondiente al extremo inferior del rango.

Nivel	**Rango de XP**
Novato	0-49
Principiante	50-149
Competente	150-249
Avanzado	250-399
Experto	400-550

Cuadro 4.6: Correspondencia entre nivel de usuario y rango de XP asociado. Se utiliza una escala de rangos con una progresión no lineal [230].

La actualización de XP para los usuarios se da de la siguiente manera:

- Las clasificaciones erróneas, en la implementación actual no restan puntos: si un usuario no genera una clasificación correcta, mantiene el XP con el que contaba antes de emitir una clasificación (es decir, que corresponde a la suma de 0 XP).
- Cada etiqueta correcta corresponde suma 5 XP.
- 5 etiquetas emitidas en la instancia de clasificación suman 5 XP.

4.3.3.3.2. Generación de etiqueta consenso

El objetivo final del etiquetado de variantes, es que el aporte de los usuarios a través de estas etiquetas sea tomado en el proceso posterior de clasificación, en el entrenamiento de modelos de aprendizaje automático supervisado. Para un gran número de variantes no se cuenta con una asignación de una clase de significado clínico o con un *gold standard*, por lo que en el presente trabajo se platea la posibilidad de que parte tanto del aprendizaje de los modelos como de la evaluación, se realicen usando etiquetas subjetivas de un conjunto de usuarios que emiten etiquetas, y que en su mayoría pueden ser no expertos. El hecho de que las etiquetas son brindadas por múltiples usuarios implica que diferentes usuarios pueden brindar diferentes etiquetas a la misma variante, y de allí surge la necesidad de contar con un consenso a partir de dichos aportes individuales, como fue planteado en secciones anteriores. Las variantes que se consideran inicialmente para este abordaje cuentan con un conflicto en la asignación de un significado clínico proveniente desde ClinVar, y el objetivo final es que desde la plataforma se cuente con un consenso diferente a dicho conflicto. Si

[8] *Experience Points, en inglés.*

bien una posibilidad es que el conflicto se mantenga, y el proceso de generación de consenso continúe, valiéndose de la madurez progresiva de los usuarios en la tarea, es necesario contar con una estrategia para la generación de dicho consenso.
El problema de obtener una etiqueta consenso puede identificarse como un problema de clasificación del tipo *Learning from crowds* o etiquetado a partir de multitudes, es decir, el proceso de agregar las etiquetas que fueron asignadas por múltiples usuarios o anotadores en una única etiqueta. Como fue descrito en el Capítulo 2, existen varios métodos ya estudiados para aprender una etiqueta consenso a partir de multitudes, y la elección del método depende de la tarea a realizar y de la disponibilidad de las puntuaciones de fiabilidad de los anotadores.

Abordaje inicial

Las variantes tomadas con conflicto y significado incierto de ClinVar (que son las que participan inicialmente en la etapa de etiquetado) provienen de un sistema que no genera un consenso en caso de contar con distintas etiquetas para la misma variante. ClinVar en este sentido, es un archivo para brindar un significado clínico a variantes mediante los usuarios de la misma. Cada variante cuenta con una o múltiples etiquetas, y estados de revisión, que pueden ser considerados como niveles de confianza en las aserciones realizadas (dado que indican el nivel de soporte con el que cuenta un reporte realizado). Si múltiples entidades reportan valores distinto de significado clínico para la misma variante, ClinVar reporta que hubo un conflicto (etiqueta final), y se listan todos los valores individuales reportados. Además, en estos casos el estado de revisión de la variante se represente con 0/4 estrellas a modo de indicar la falta de certeza en la interpretación realizada. ClinVar no tiene mecanismos para resolver el conflicto, y la única estrategia implementada es mostrar el significado clínico asociado al reporte que tiene un *expert panel* que lo sustenta (el estado de revisión más alto). En este último caso, el conflicto no es reportado.
En el presente trabajo se comenzó planteando una abordaje inicial para la obtención de un consenso, que parte del abordaje de ClinVar al momento de generar conflictos. Dados los usuarios con los que se cuenta en una versión inicial y controlada de la plataforma, se puede partir de un escenario en el que la etiqueta de un experto sea tomada como una etiqueta consenso, en caso de contar con ella, dado que el nivel de experto se encuentra ocupado por profesionales que se han desempeñado en la tarea por un tiempo muy prolongado. En este sentido, tal como ClinVar, si una variante cuenta con una serie de etiquetas asociadas, y una de ellas corresponde a un experto, entonces el consenso será la etiqueta brindada por el experto. En caso de que dicho escenario se de con dos expertos que difieren en la etiqueta, se emite un conflicto. Si los expertos son más de dos, se elige un abordaje de voto de mayoría entre ellos; si dicha estrategia no puede ser aplicada (por ejemplo, hay un voto o cantidades iguales de voto a cada etiqueta), se emite un conflicto. Si no hay expertos entre los anotadores, entonces se pasará a dar prioridad a Avanzados y Competentes, tomando su consenso por sobre el de principiantes y novatos, y siguiendo la estrategia de máximo voto mencionada anteriormente. Por último, se considera un consenso de Principiantes y Novatos, en este caso, aplicándose el máximo voto directamente. En caso de que este no pueda emitirse por contar con un voto por etiqueta, entonces se mantiene el conflicto. En cada caso que se actualice la etiqueta consenso, se genera un estimativo de nivel de confianza, a los efectos de tomarlo como insumo en la clasificación automática. Si la etiqueta consenso se genera con Expertos, el nivel de confianza es 1, si se genera con Avanzados y/o Competentes es de 0.7, si se genera con Principiantes y/o novatos es de 0.3. Todas las variantes cuya etiqueta consenso que se mantenga en un conflicto, seguirán el flujo del resto de los datos que pertenecen a la etapa de etiquetado: vuelven a sugerirse a los usuarios, con el fin de lograr emitir un consenso a futuro.

Cabe destacar que este abordaje no cubre la complejidad de la tarea planteada ni del aspecto biológico subyacente, y fue planteado como una estrategia inicial para generar un consenso a partir de herramientas con las que ya se cuente. Se pretende que a medida que se haga la recolección de datos se explorarán herramientas y métodos más adaptados al problema.

Uso de herramientas externas

Como segundo abordaje implicó el uso de herramientas que permitan el agregado de etiquetas a los efectos de obtener un consenso, a través de diferentes metodologías. En este sentido algunas de ellas son:

- CROWDLAB (Classifier Refinement Of croWDsourced LABels): es una herramienta que permite levar a cabo una serie de tareas relacionadas con el acondicionamiento de etiquetas obtenidas de varias fuentes. Se puede generar una etiqueta de consenso para cada ejemplo que agregue las anotaciones disponibles; un *score* de confianza para la probabilidad de que cada etiqueta de consenso sea correcta; un *score* para cada anotador que cuantifique la corrección general de sus etiquetas [231]. Para implementarlo se utiliza el paquete `cleanlab` en Python.
- Crowd-Kit: también es una librería de Python que implementa métodos de agregación para la anotación de varias fuentes y ofrece las métricas relacionadas. Esta herramienta, a diferencia de CROWDLAB se basa en métodos basados en los modelos más tradicionales para la generación de consensos (como por ejemplo DS). Con ella se pueden abordar: implementación de métodos de agregación de uso común para etiquetas categóricas, textuales y de segmentación, y métricas de incertidumbre, coherencia y concordancia con el agregado [232].

Estas herramientas fueron puestas a prueba con datos de ejemplo a los efectos de evaluar su pertinencia en el contexto del trabajo actual, sin embargo la implementación final de la herramienta no las integra. Al igual que fue mencionado anteriormente, se espera que a medida que se recolecten datos, los distintos enfoques se puedan poner a prueba.

4.4. Resultados preliminares y discusiones

4.4.1. Disponibilidad de la aplicación

La plataforma en su versión MVP se encuentra disponible a través de la web y corriendo satisfactoriamente en un servidor perteneciente al Grupo de Ingeniería Biológica del CENUR Litoral Norte (sede Paysandú) [233]. El servidor cuenta con las siguientes especificaciones: Intel(R) Core(TM) i7-3770 CPU 3.40GHz, 15GB de RAM. El equipo cuenta con sistema operativo Linux, distribución Ubuntu 20.04.4.
La plataforma web es de acceso público mediante previo registro, a través del cual los usuarios ya pueden acceder a las prestaciones de la misma.
El código del proyecto en su versión de desarrollo más reciente se encuentra disponible en GitLab [234]. El proyecto de la aplicación [234] consiste en dos grandes partes, agrupadas en directorios dentro del repositorio:

- Aplicación web (`app/`): aplicación web Dash app multipágina con login.
- Servidor de aplicación web (`app_server/`): Servidor API Flask con servicios web y persistencia en MySQL.

Además, el repositorio cuenta con un directorio en donde se almacena la base de datos (`DB/`). La base de datos contiene archivos grandes, por lo que su gestión se hace a través de git LFS[9]. El repositorio cuenta con las instrucciones necesarias para la instalación, tanto de la API como de la aplicación web, a través de la cual se puede, tanto correr la aplicación en el navegador propio en modo desarrollo local, como correr la aplicación en modo producción usando Gunicorn [235].

[9] *Large File Storage, en inglés.*

4.4.2. Estado actual

La plataforma actualmente se encuentra disponible, no obstante no cuenta con una actividad generada, más allá de las pruebas realizadas en la etapa de desarrollo. Es por ello que tampoco se cuenta con un conjunto de datos válidos para su uso a posteriori, ni se ha podido evaluar el desempeño de las estrategias de aprendizaje o de etiquetado. Se espera que a partir de la etapa actual de finalización de la implementación, se comiencen a generar datos, e inicie una nueva instancia de evaluación. Como consecuencia, las tablas de clasificación y entrenamiento comenzarán a completarse y generar insumos.
Actualmente el sistema cuenta con 15 usuarios autenticados y prontos para iniciar su actividad, los cuales pertenecen o se encuentran relacionados al grupo de trabajo en el que se enmarca la tesis y a instituciones que colaboran con el mismo (y viceversa). Entre esos 15 usuarios, hay 3 expertos, 3 avanzados, 1 competente, 7 principiantes y 1 novato. Todos los usuarios pertenecen a instituciones locales, y se encuentran distribuidos entre: Facultad de Medicina, principalmente Departamento de Genética (FMed) , Institut Pasteur de Montevideo (IPMon) y el Grupo de Ingeniería Biológica del CENUR LN (IngBio). Dos de los usuarios actuales son administradores, y cinco en total participaron tanto en el planteo y toma de decisiones en el desarrollo de la plataforma, como en el armado de la infraestructura (2 expertos, 1 avanzado, 1 principiante, 1 novato). Actualmente hay 2 usuarios no administradores que cuentan con el nivel superior de permisos, es decir, que pueden acceder a analizar datos de pacientes. En la Tabla 4.7 se muestra una distribución de los usuarios autenticados en la plataforma de acuerdo a la institución a la que pertenecen y los niveles de pericia que tienen. Además se hace un resumen de la participación de los usuarios en el desarrollo de la plataforma.

Institución	**Nivel de pericia**					**Total**	**Nivel de participación**		
	Novato	**Principiante**	**Competente**	**Avanzado**	**Experto**		**Ninguno**	**Diseño**	**Desarrollo**
FMed	0	5	1	2	2	**10**	9	1	0
IPMon	0	0	1	1	1	**3**	1	2	0
IngBio	1	1	0	0	0	**2**	0	1	2
Total	1	6	2	3	2	**15**			

Cuadro 4.7: Distribución de los 15 usuarios autenticados en la plataforma en la actualidad de acuerdo a la institución a la que pertenecen y los niveles de pericia que tienen. En la parte derecha se realiza un resumen del nivel de participación que tuvieron los distintos usuarios que se encuentran autenticados en el desarrollo del proyecto: 5 de los usuarios que se encuentran autenticados participaron directamente, algunos en la toma de decisiones y diseño, otros en el desarrollo y puesta a punto de la infraestructura, y algunos en ambos casos.

Actualmente la plataforma se sirve mayoritariamente de repositorios de datos públicos y herramientas gratuitas, tanto para la obtención de variantes, como la información que las complementa. El único recurso que no es público, es URUGENOMES, que en este caso aporta las variantes de algunos pacientes con enfermedades raras a modo de prueba. Dado que se trata de datos sensibles, los mismos no se encuentran disponibles para su uso por fuera de la órbita de IPMon.
En la Tabla 4.8 se presenta un resumen de la cantidad de variantes de ClinVar con las que se cuenta en la base de datos, distribuidas según su uso y tipo. ClinVar (un subconjunto de la versión diciembre 2022, al momento de esta documentación) se encuentra aportando `460451` variantes en total: `352674` de ellas aportan a la etapa de etiquetado y `107777` a la etapa de aprendizaje.
La base de datos no contiene todas las variantes disponibles en ClinVar, dado el procesamiento que han sufrido, el cual fue descrito anteriormente. A su vez, la plataforma no carga en una sesión cada vez todas las variantes disponibles en la base de datos. Esto se hace con dos objetivos: balance de clases, y control de tiempos de latencia. Respecto al balance de clases, en la tabla 4.8 se puede observar como las variantes de significado incierto son la clase dominante en cuanto a cantidad respecto a los otros tipos. Ese aspecto refleja, no solo la distribución de datos en ClinVar, sino también la naturaleza de la dificultad en la tarea de determinación de significado de patogenicidad. Por ello, alos efectos de mantener un balance entre los tipos de variantes que se le presentan al usuario, se tiende a

reducir la cantidad de variantes VUS al momento de cargar los datos a la plataforma. En este sentido, por ejemplo, en la etapa de etiquetado la cantidad de variantes VUS cargada son 50000, con el fin de equilibrar con las variantes conflictivas. Respecto a los tiempos de latencia, cabe destacar que al momento de presentar la vista de tabla en la selección inicial de variantes, se cargan en memoria todas las variantes pertenecientes a la clase correspondiente. En este sentido, dado lo demandante del proceso, dependiendo de la cantidad de variantes la página puede tardar varios segundos en cargar la tabla completa. En el caso de la etapa de etiquetado, donde se cuenta con más de 350000 variantes, puede superar los 10 segundos de espera. Más allá de que la estrategia de presentación de las variantes no es la adecuada, temporalmente la obtención de un sub-conjunto de las variantes totales, también aporta en la reducción de tiempos.

Tipo de variante	Etapa de etiquetado	Etapa de entrenamiento
Benigna	-	18515
Probablemente Benigna	-	22741
Probablemente Patogénica	-	2607
Patogénica	-	9952
VUS	309702	51789
Conflicto	42972	-
Factor de riesgo	-	375
Respuesta a drogas	-	1798
Total	**352674**	**107777**

Cuadro 4.8: Distribución de variantes de ClinVar disponibles actualmente en la base de datos para servir a la plataforma según su tipo y la etapa en la que se las utiliza. La base de datos que contiene el conjunto de datos totales se encuentra alojada en un servidor perteneciente al Grupo de Ingeniería Biológica, del CENUR Litoral Norte. La plataforma funciona con un sub-conjunto de las variantes totales, el cual consiste principalmente en el conjunto total de variantes y un sub-conjunto de variantes de significado incierto.

Respecto a variantes de pacientes, actualmente se cuenta con un número reducido de variantes pertenecientes a datos de WGS de dos individuos participantes de la fase 3 del proyecto URUGENOMES, los cuales se utilizan en pruebas internas. Atendiendo a las posibilidades de diagnóstico, los datos, además de ser sometidos a la anotación, fueron filtrados de acuerdo a los indicios preliminares correspondientes a cada casi de estudio. En base a esto, se cuenta, para uno de los pacientes con un sub-conjunto de 520 variantes exónicas no sinónimas que se encuentran en la muestra en heterocigosis con menos de un 1 % de frecuencia poblacional máxima. Para el segundo paciente se cuenta con un sub-conjunto de 520 variantes exónicas no sinónimas que se encuentran en la muestra en heterocigosis con menos de un 0.5 % de frecuencia poblacional máxima. Dichos datos no se encuentran disponibles ni en el repositorio, ni en la versión de desarrollo alojada fuera de la órbita del Institut Pasteur, dada la privacidad de los datos. Los datos de pacientes disponibles en la versión actual consisten en datos de prueba.

4.4.3. Evaluación general de la funcionalidad

La plataforma final implementada presenta la capacidad de:

- Permitir la autenticación y registro de usuarios, posibilitando seguir el progreso de los usuarios y de las instituciones a las que pertenecen a través de los mismos. Además, contar con los datos que los usuarios permite seguir el proceso de clasificación al que están expuestas las variantes.n
- Evaluar variantes en abstracto y en el contexto de una muestra específica (cuando se cuente con la información) para asociar la evidencia pertinente con la clasificación de una variante.

- Acceder a la información de una variante obtenida a partir de la anotación disponible, organizada y categorizada en una interfaz.
- Permitir al usuario entrenarse mediante estrategias básicas de gamificación en el proceso de valoración de una variante.
- Permitir la generación de etiquetas provenientes de usuarios locales con distintos niveles de pericia, y la generación de un consenso a través de estrategias preliminares.
- Permitir que los usuarios ingresen nuevas variantes al sistema para su posterior evaluación tanto por ellos mismos como por el resto de los usuarios que están participando en los procesos de la plataforma.
- Permitir un abordaje inicial de predicción de patogenicidad de una variante ingresada a través de un algoritmo entrenado y seleccionado en el marco del mismo trabajo.

La funcionalidad de la plataforma se presenta a través de videos que muestran los distintos niveles de actividades que permite la misma [236]. Como puede observarse, la implementación de los casos de uso principales se encuentra completa y funcional. Los tiempos de respuesta implicados en cada tarea se presentan de forma aproximada y resumida en la Tabla 4.9.

Tarea		**Tiempo de respuesta de la página (ms)**
Login a Home		247
Entrenamiento	Home a Entrenamiento	1500
	Ver detalle de variante seleccionada	**4100**
	Ver detalle de variante random	**4100**
	Cambio de pestaña	**190**
	Emisión de etiqueta a obtención de feedback	1500
Etiquetado	Home a Etiquetado	1500
	Selección inicial de tipo de variante a comienzo	**6800**
	Ver detalle de variante seleccionada	**5100**
	Ver detalle de variante random	**5100**
	Cambio de pestaña	190
	Emisión de etiqueta a obtención de feedback	1500
Entrenamiento a etiquetado		1900
Etiquetado a entrenamiento		1900
Logout		**320**

Cuadro 4.9: Tiempos de respuesta de la página ante las tareas involucradas en los casos de uso principales: entrenamiento de usuarios y etiquetado de variantes. El tiempo se midió como aquél que transcurre entre una acción de un usuario en la interfaz, y el momento en el que se da una respuesta en la interfaz, no representa tiempos de respuesta de la API.

Los tiempos medidos representan el intervalo entre la emisión de una acción por parte de un usuario, y la respuesta de la interfaz web completamente cargada. Como puede observarse, los tiempos de respuesta son elevados respecto a lo esperado usualmente. Se esperaría que un tiempo de respuesta aceptable se encuentre entre los 200 milisegundos y 1 segundo, en el cual el usuario probablemente no note el *delay*. Incluso, para mejor experiencia de usuario se considera que por debajo de los 200

milisegundos se tendría una percepción de respuesta inmediata. En este sentido, considerando que cualquier respuesta que supere el segundo de espera puede ser problemática, provocando un posible desestímulo a usar la herramienta, la plataforma implementada tiene un importante problema a atacar en el futuro.
Como puede observarse en la Tabla 4.9 los tiempos más complejos en *performance* son aquellos en los que las tareas dependen o de grandes cantidades de datos, o de cantidades grandes de salidas y recursos gráficos. Tanto en la etapa de entrenamiento como en la de etiquetado se identifica el mismo patrón de demanda en la vista del detalle de una variante, ya sea mediante selección en una tabla o de forma aleatoria. Más allá de que en uno de los casos el tiempo es mayor, ambas tareas tienen en común la mayor parte de la vista de la variante. El proceso de vista de la variante implica que los *callbacks* que manejan esta interactividad tengan una cantidad de salidas muy grande, con múltiples llamadas a la API. Teniendo en cuenta que una de las principales limitaciones de rendimiento de las aplicaciones Dash son los *callbacks*, sumado a las ineficiencias propias del código implementado, es acorde que se haya obtenido como resultado la limitación en los tiempos mencionada [237]. Además de la demanda en *callbacks*, mostrar la información de una variante en distintas pestañas implica renderizar gráficos en Plotly. Si bien la cantidad de datos que se grafican es reducida, la demanda en la renderización tiene un aporte importante en el tiempo total de cargado de la página. De hecho, se ha comprobado cómo la incorporación de gráficas de frecuencia aumenta el tiempo de cargado en orden de segundos. Se considera que el porte de la plataforma que se busca implementar requiere de herramientas distintas a Dash para la implementación de la interfaz. Más allá de ello, se puede plantear como trabajo futuro la exploración de distintas estrategias planteadas para mejorar el desempeño de los *callbacks* y la renderización de gráficas, como las mencionadas en [237]. Además de los tiempos demandantes de la presentación del detalle de las variantes, se destaca la tarea a partir de la cual se pasa a la etapa de selección y vista de la variante. Esta tarea, la cual se encuentra únicamente en la etapa de etiquetado, implica oprimir un botón de "comenzar", luego de que se selecciona el tipo de variante con el que se trabajará, o se ingresa un archivo. Dicho proceso tarda actualmente casi 7 segundos, realizando la prueba con la selección de variantes del tipo *default*. Recordando la distribución de variantes entre etapas, las variantes por *default* consisten en variantes Conflictivas y VUS provenientes de ClinVar. Considerando el muestreo de variantes VUS mencionado anteriormente, generar la tabla para la selección de estas variantes, implica que se carguen en memoria más de 90000 filas. Si bien es una cantidad reducida respecto a la cantidad de variantes que se pueden adquirir, la forma en la que se muestran las variantes resulta poco eficiente para la cantidad actual. Al momento de realizar la prueba seleccionando las variantes de pacientes, el tiempo que transcurre hasta que se presenta la tabla se reduce a menos de la mitad que el tiempo en el que se muestran las variantes por defecto. La dimensión de los datos también se refleja en las interacciones más pequeñas, como la selección en desplegables, que en general llevan 200 milisegundos más que en otros caso. Cabe destacar, que la etapa de entrenamiento también contiene la vista de tabla para la selección de variantes, sin embargo, la separación de conjuntos de variantes dependiendo de la dificultad adaptada a cada nivel de usuario, hace que dicha tabla se genere sobre un conjunto menor de datos. Si bien la selección de variantes en una tabla resulta cómodo y adecuado con conjuntos de variantes reducidos, es un mal enfoque para un contexto donde el volumen de variantes sea mayor, por lo que nuevas estrategias serán requeridas en la presentación.

Entrenamiento de usuarios

En lo que respecta al entrenamiento de usuarios, los resultados preliminares de la implementación muestran un flujo completo de trabajo desde que se ingresa a la plataforma, hasta la emisión de una etiqueta obteniendo el *feedback* correspondiente. La implementación de una versión inicial de una instancia de aprendizaje permitió identificar los aspectos básicos implicados en la tarea, sin embargo, hay varios aspectos a atacar en versiones posteriores. Los dos grandes puntos que merecen un mayor desarrollo son el sistema de gamificación, y el enfoque del aprendizaje en las reglas ACMG/AMP.

Si bien no se cuentan con resultados respecto al desempeño, se puede mencionar que como abordaje preliminar, el sistema de niveles y puntos que se implementa de momento no alcanza a reflejar la complejidad de la tarea, y del aprendizaje que se quiere lograr. Si bien se incorporaron algunos elementos de diseño de juegos que han extendido su uso en contextos de aprendizaje o entornos no lúdicos, se pretende que a futuro la plataforma utilice ampliamente herramientas de gamificación en la instancia de aprendizaje. Hasta ahora se ha implementado una versión preliminar que intenta incluir un sistema de puntos y distintos niveles, y a esto se le puede incorporar otros elementos como "premios", intercambio entre usuarios, un *feedback* más amplio, visualización más estimulante para mostrar barras de progreso y paneles, etc. Además, hay aspectos intrínsecos a la tarea de clasificar una variante como tal que deben ser considerados también. En el sentido de adicionar cambios que pueden aportar una riqueza diferencial a esta instancia del sistema, se listan algunas ideas a continuación:

- La tarea de aprendizaje debería ser compartimentalizada, de modo que se reconozcan puntos estratégicos en el aprendizaje de las reglas a conocer. Esto permitiría que el aprendizaje se pueda hacer gradual, atacando cada aspecto por separado cada vez, además de adaptar el aprendizaje a los puntos que más le cuestan al usuario. En la implementación actual, no se logra reconocer en qué es lo que el usuario puede estar fallando al momento de no acertar una etiqueta. Reconocer esto puede ayudar a que el proceso general sea más estandarizado.
- Relacionado con el ítem anterior, al diferenciar las tareas, se pueden asignar distintos niveles de XP según su complejidad, tanto pre-establecida, como para el usuario. Por ejemplo, si es una tarea que se ha demostrado que le presenta una gran dificultad al usuario, se puede premiar su logro con más XP que en la normalidad.
- Eventualmente, la estrategia de *spaced repetition* puede sustituirse por la inferencia de estado de conocimientos y la dinámica de la memoria en determinada destreza o combinación de varias destrezas [238].
- Brindar una mayor asistencia al usuario en el proceso de aprendizaje, a través de guías, mayor interactividad, información complementaria, y *feedback* [239].
- Realizar un sistema de puntos que sea más adaptado al problema, incluyendo la posibilidad de que el usuario pueda bajar de nivel eventualmente, y una interacción más cercana con la instancia de etiquetado mediante la cual se pueda modelar cómo la misma aporta en la obtención de puntos.
- Además de que a través de los XP se dificulte más llegar a niveles superiores, debería buscarse una estrategia para que también se dificulte mantener los niveles más altos, o aquellos a través de los cuales se llega a los niveles más altos (por ejemplo, Competente). En este sentido, se podría tener en cuenta la actividad temporal del usuario, de modo que se requiera una actividad más frecuente para mantener el nivel en el que el usuario se encuentra.

Respecto al desarrollo de estrategias que permitan acercar al usuario al conocimiento de las reglas ACMG/AMP, se considera que un buen objetivo a futuro es que la plataforma sea un espacio centralizado que logre acercar la adopción generalizada de estándares para la curación de variantes a la práctica clínica, atención sanitaria, etc. En este sentido, se requiere una mayor formalización de la tarea de enseñanza de estas reglas, independientemente de que se lo quiera lograr a través de herramientas de gamificación. Para ello, primero es necesario generar una base de material estandarizado, y luego tratar de adaptarlo en la interfaz, a los efectos de que el progreso del usuario en la evaluación interactúe con la evaluación de las reglas. Un ejemplo de un posible enfoque seguir es el de la interfaz

de curación de ClinGen (VCI[10]), implementada en forma de plataforma de clasificación que permite la aplicación de los criterios de las guías ACMG/AMP en la evidencia [201]. Dicha plataforma, además de ofrecer una vista de evidencia de la variante, que recolecta la información relacionada a la misma, ofrece una vista de interpretación que permite al usuario interactuar con las reglas. Otras plataformas como Varsome [89] o Franklin [91] ofrecen un veredicto de las reglas cumplidas, pero permiten al usuario interactuar con las mismas, e identificar cómo se modifica la clasificación con cada cambio realizado. Este enfoque sería interesante de explorar, en un escenario en que el usuario pueda aprender cómo se combinan las distintas reglas. Un aspecto no menor respecto a las últimas plataformas mencionadas, es que las mismas cuentan con una implementación propia de las reglas ACMG/AMP, las cuales integran la información brindada por los usuarios también, Esto brinda un mayor margen de maniobra comparado con el uso de reglas brindadas por una herramienta externa, la cual cabe destacar, requiere de actualizaciones. Actualmente, el *feedback* de las reglas se brinda mediante la integración de Intervar a la anotación, con una versión que para muchas variantes ha quedado obsoleta, por lo que implementaciones futuras van a requerir que la fuente brindada respecto al cumplimiento de las guías sea confiable.

Etiquetado de variantes

En lo que respecta al etiquetado de variantes, los resultados preliminares también muestran que se puede realizar un flujo completo desde la selección de una variante hasta la emisión de una etiqueta. Al igual que en la etapa de etiquetado, no se cuenta con una recolección de datos que permita identificar patrones o realizar un análisis más profundo del desempeño de la tarea, obteniendo como resultado actual, la implementación de la funcionalidad. Los elementos básicos requeridos para tener una versión inicial del etiquetado funcionando se encuentran dados, sin embargo se requerirá dar una mayor desarrollo a la generación de consensos y poner atención sobre la calidad de los datos que se brindan para etiquetar.

En cuanto a la generación de etiquetas consenso, más allá de la posibilidad de usar herramientas que integren elementos que en la implementación actual no son cubiertos, y son de gran importancia (como medidas de concordancia, medidas de confianza, etc.), es importante que a futuro se tengan en cuenta las características intrínsecas del problema para contar con un modelo propio. Tratar con la baja fiabilidad que caracteriza a los datos recogidos de un conjunto de expertos es un reto a futuro en este trabajo. Más allá de que se comparten muchas características con otros problemas donde el etiquetado a través de múltiples anotadores está presente, la clasificación de variantes y la adquisición de experiencia en la tarea no están tan estandarizados como otras, más allá de los esfuerzos generados por establecer estándares. A continuación se listan una serie de aspectos que se consideran importantes a re-evaluar y tener en cuenta en implementaciones posteriores:

- En el presente trabajo se parte de un escenario en el que la fiabilidad de los anotadores no se conoce, pero se puede "medir" de forma directa indirecta, dado el nivel de pericia con el que cuentan. En el contexto de la plataforma objetivo, este nivel de pericia también estaría directamente relacionado con el desempeño del usuario clasificando variante (en cualquiera de los dos modos). La implementación actual no cuenta con un sistema validado de actualización de los niveles de pericia de los usuarios, y la asociación de niveles de pericia iniciales son establecidos según la evaluación de un experto (en el nivel más alto). Más allá de esto, sería importante contar con una fuente robusta de actualización de estos niveles.
- El origen de las diferencias en la clasificación puede ser diverso, y puede ir desde diferencias en la experiencia o conocimiento con los que cuenta cada usuario al momento de emitir una etiqueta, hasta la información disponible para realizar el etiquetado. Si bien esta última debería

[10]Variant Curation Interface, en inglés.

ser la misma, el hecho de que los expertos realicen la etiqueta en distintas instancias de tiempo, la información disponible para la variante puede cambiar tras actualizaciones de la base de datos. Sería de importancia trabajar en el origen de estas diferencias, y aplicar cambios que son directamente reconocibles, como formas de automatizar la re-evaluación de variantes por, en un principio, los mismos usuarios que ya la han evaluado, cuando la información de las mismas ha sido actualizada en alguna de sus características. Esto tendría como consecuencia el diferenciar no solo entre evaluadores distintos, sino también considerar que un mismo evaluador asigne etiquetas a la misma variante en distintas instancias de tiempo.

- Se puede calcular el acuerdo inter-evaluadores pero no se puede calcular la fiabilidad respecto a la etiqueta correcta de la variante, porque el consenso es desconocido en este caso. Lo que si se puede tener en cuenta es la fiabilidad inter-evaluadores, en función de la consistencia en la escala usada al momento de clasificar la variante [240]. Para esto sería relevante buscar una escala acorde, para poder cuantificar distintos niveles de discrepancia entre usuarios (por ejemplo, no debería tomarse igual una discrepancia entre las etiquetas "Patogénica" y "Probablemente Patogénica", que entre "Patogénica" y "Benigna", tratándose de anotadores en el mismo nivel).

- También se podría tratar de identificar posibles correlaciones entre las etiquetas de distintos expertos. Hay trabajos que a través de la autoconsistencia de las etiquetas proporcionadas por un experto han podido identificar a los etiquetadores que introducen ruido aleatorio, el tipo de ruido menos perjudicial [241, 242].

- Teniendo en cuenta que la información de las variantes puede ir cambiando, y eso por ende puede cambiar la etiqueta brindada, se podrían generar distintas instancias de las variantes etiquetadas, a los efectos de poder hacer evaluaciones a futuro sobre el comportamiento de los usuarios frente a la información, y cómo los cambios van moldeando el mismo.

- Se puede trabajar en el modelado de las fuentes de ruido que podrían perjudicar la calidad de las etiquetas proporcionadas, basándose en la literatura relacionada [242].

El destino final de los datos generados en esta instancia será la clasificación automática de variantes, por lo que es necesario que los datos etiquetados cuenten con una etiqueta consenso que sea lo más robusta posible. Además, los datos que se brinden a etiquetar deben ser informativos y lo más completos posibles, que no den lugar a posibles ambiguedades que se generen por la calidad de los datos y no su complejidad intrínseca. En este sentido, la información brindada al usuario a partir de la cual hace su clasificación es en la mayoría de los casos puede resultar insuficiente, por lo que se requerirá um mayor trabajo en el curado de los datos que se brindan para etiquetar. La importancia de obtener posibilidades de conseguir fuentes distintas de datos respecto a la anotación, también se evidencia en esta instancia, dado que la desactualización en las bases de datos de las cuales se sirve la anotación puede derivar en veredictos incorrectos.

4.4.3.0.1. Servicio de clasificación de usuarios

Este caso de uso en la actualidad se encuentra poco desarrollado, alcanzando una etapa básica en la que el usuario puede ingresar un archivo, y se le brinda una predicción. Más allá de los aspectos que puedan complementar esta versión ya implementada, un aspecto que no fue explorado y sobre el cual se plantea trabajar a futuro es un panel de expertos como servicio. Esto se refiere a que no solo se brinde como servicio el veredicto de una herramienta de clasificación automática, sino también de los usuarios que participan en la plataforma. Siguiendo el enfoque de ClinGen, un conjunto de expertos puede ser seleccionado para realizar curaciones finales respecto a etiquetas brindadas otros usuarios de la plataforma. Tomando nuevamente como ejemplo a Varsome o Franklin, cobraría sentido contar

con una plataforma que genere un sentido de comunidad entre los anotadores. Esto permitiría no solamente que determinados usuarios puedan nutrirse del aporte de otros expertos, sino también que el grupo de expertos que etiquetan juntos como una filiación genere intercambio de conocimientos y experiencias.

4.4.3.0.2. Clasificación de variantes de pacientes

Como ya fue mencionado, la funcionalidad de la plataforma como interpretador de variantes de una o varias muestras no era el objetivo inicial, y se derivó en la misma como un efecto secundario. Las operaciones básicas para que este aspecto se encuentre funcional con una experiencia de usuario mínima no se encuentran del todo dadas. Por ejemplo, el usuario no tiene medios en la actualidad para almacenar el proceso que se va generando con un paciente, la cual es una funcionalidad de importancia a desarrollar. Cuando se evalúa un conjunto de variantes, es importante reconocer cuáles ya han sido evaluadas y descartadas. No obstante, esto no quiere decir que tras el ingreso de un conjunto filtrado de variantes, no se pueda llevar a cabo una evaluación, las funcionalidades mínimas están dadas para la tarea de forma poco eficiente: selección del paciente, filtrado de variantes de interés mediante una tabla, y descarga de la variante de interés una vez que se la encuentra. Además de la poca flexibilización en la tarea, actualmente no están dados los medios para que el usuario ingrese sus propias variantes de pacientes, para lo cual tiene que participar un administrador si o si. Más allá de que la implementación de este aspecto no se encuentre con una funcionalidad completa deseada, se considera que en algunos casos, la experiencia de visualizar una variante mediante una interfaz puede mejorar respecto a otros mecanismos de evaluación, por ejemplo, planillas.

4.4.3.0.3. Ingreso de nuevas variantes

El ingreso de nuevas variantes a la plataforma, si bien se encuentra implementado en la actualidad, no cuenta con las características que se buscaría para una implementación funcional y cómoda para el usuario. La forma en la que se ingresan las nuevas variantes es mediante una planilla, la cual debe contar con una determinada cantidad de columnas, y un orden específico de las mismas. Esta implementación preliminar resulta directa al momento de incluir variantes nuevas en la base de datos, pero no así para el usuario que necesita manipular los datos previamente a su ingreso. Esto implica no solo que la variante sea anotada previamente, sino que en lo posible, dicha anotación se haya hecho con las bases de datos específicamente requeridas, y que las columnas se hayan ordenado intencionalmente. Esto implica que actualmente el ingreso de variantes nuevas no pueda realizarse sin asistencia o sin manipulación de herramientas de análisis de datos y/o anotación. Si bien se brinda un *template* para que el usuario logre reconocer la estructura de la tabla o usarlo, completar la información requerido sobre el mismo puede ser trabajoso dada la cantidad de columnas, y más aún si se considera la posibilidad de que se ingrese una gran cantidad de variantes nuevas.

4.4.4. Discusión general

Como puede observarse y como ya fue mencionado, la implementación actual se encuentra en una etapa meramente preliminar, la cual permitió sentar bases necesarias para brindar funcionamiento al sistema. Más allá de contar con las bases para iniciar pruebas, la plataforma aún no cuenta con una evaluación de experiencia de usuario formal. No obstante, en el Apéndice C, se presentan una serie de experimentos diseñados para dicha evaluación. Además, en el Apéndice C se presenta una evaluación cualitativa realizada por usuarios que forman parte y cercanos al grupo de trabajo. A pesar de que esta evaluación no cuenta con la rigurosidad requerida para hacer una discusión acorde (dados los sesgos presentes y su implementación), se pueden obtener algunos putos de vistas preliminares respecto a su uso y posibilidades brindadas. En general, en función de las evaluaciones realizadas

se podría decir que la plataforma cuenta con potencial en su uso esperado. Se ha valorado positivamente la forma en la que se se concentra y agrupa la información relevante de distintas fuentes, necesaria para una clasificación. Dependiendo del tipo de usuario que realizó la evaluación, se logran identificar distintos aspectos a desarrollar a futuro, los cuales se centran principalmente en ampliar funcionalidades para mejorar el aprendizaje, y brindar un manejo más fluido para la evaluación de variantes ingresadas por el propio usuario.

Más allá de los aspectos puntuales evaluados en función a cada caso de uso, hay aspectos transversales que afectan todas las instancias. Uno de ellos es la información brindada al usuario, a partir de la cual se genera la emisión de una clasificación. Esta información se considera adecuada en la instancia actual, pero para implementaciones futuras se requerirá trabajar más en detalle, no solo en la calidad de la misma, sino en los aspectos que se muestran. Por un lado, se ha podido detectar que en algunos casos, la información presentada para la variante puede diferir de la información encontrada en los medios oficiales, y esto se debe a la falta de actualización en los medios elegidos para complementar con información las variantes. Esto actualmente se realiza a través de la herramienta ANNOVAR, utilizada de forma gratuita, la cual no cuenta con el mismo nivel de actualización para todas las bases de datos de las cuales se nutre. En este sentido, brindar información desactualizada al usuario, en el contexto en el que la nueva evidencia juega un rol fundamental, es un problema a mejorar con urgencia. Como propuesta a futuro se plantea ampliar el trabajo en la integración de datos, basado en el uso de APIs de las bases de datos individuales. Esto podría ir acompañado de un cambio en la estrategia de almacenamiento de datos elegida.
Relacionado con el aspecto mencionado anteriormente, las bases de datos más actuales se encuentran relacionadas a un ensamblaje de referencia más actual al Hg37 (Hg38), por lo que una mejora que será requerida es la ampliación de los datos de la plataforma a la nueva referencia de forma completa, o a contar con ambas posibilidades de versiones. Este último caso doblaría el número de datos con los que se cuenta, por lo que se requeriría mejorar la infraestructura de almacenamiento y flujo de datos.
Además de la actualización de los datos o la referencia utilizada, como fue expresado anteriormente y como fue presentado por los usuarios, es necesario trabajar en mejorar la información brindada al usuario para realizar un veredicto, respecto a su completitud. Si bien se fundamentó la falta de por ejemplo, información relacionada con la muestra, dado el contexto del cual se obtienen las variantes, no contar con esta información puede dificultar el proceso de aprendizaje y etiquetado. En el caso del aprendizaje, puede sesgar la adopción de estándares para la clasificación en contextos exclusivamente sin una muestra, y en el caso del etiquetado, brindar etiquetas erróneas por falta de evidencia. Es por ello que mejorar el conjunto de variantes y/o la información brindada sobre las mismas es crucial. Para ello se busca a futuro poder contar con una base de datos locales, que partan de evaluaciones de pacientes propios, sobre los cuales se pueda generar una base robusta de las características que se deberían mostrar en cada caso. Además de la información de la muestra, hay otros aspectos que también pueden ser valiosos para presentar al usuario, y que actualmente no se encuentran presentes en la interfaz, como algunos datos poblacionales (números de homocigotas presentes para el alelo alternativo, etc.) y datos del desempeño de usuarios y organizaciones que participaron previamente en una evaluación, tanto de los provenientes de fuentes externas, como internos de la plataforma.
La actualización de la información de las variantes es fundamental en los casos de re-clasificación, la cual ha cobrado fuerza en los últimos tiempos, en los que la re-evaluación de una muestra puede arrojar resultados concluyentes ante un escenario nuevo de información con la que no se contaba antes. Atendiendo a esto, un aspecto que merece mayor atención es el de sistematizar la re-evaluación de variantes que ya han sido evaluadas previamente, una vez que se detecten cambios en la información disponible para las mismas. Este aspecto no está implementado, y puede ser un aporte de riqueza en la generación de consensos.
Un aspecto sobre el cual se hizo énfasis en las especificaciones fue el de internacionalización. Si bien se hizo referencia a que se buscaría primero atacar una implementación en español, a los efectos

de brindar un sentido regional a la plataforma, este aspecto resulta incompleto en la actualidad. Como puede observarse en la interfaz, el idioma de la maquinaria de la plataforma es español, sin embargo los datos que se obtienen de la base de datos, están brindados en inglés. La adaptación de qué aspectos mantener en inglés y cuáles presentar en español no fue directa, por lo que el uso de herramientas de traducción no fue una opción. Se espera que a futuro este enfoque pueda cambiar, no solo porque se espera que las fuentes de integración de datos sean otras, sino también ante la exploración de estrategias que lo permitan.
Finalmente, respecto a la interfaz gráfica implementada, la misma tiene una arquitectura de diseño sencilla y de requerimientos mínimos, pero que resulta comprensible y hace que la interacción con la aplicación sea intuitiva y accesible, más allá de las limitaciones de la tecnología mencionadas anteriormente. Las mejoras futuras de la plataforma se centrarán en mejorar la escala, el flujo de trabajo y el rendimiento, así como también en respaldar de forma continua el uso de los lineamientos estandarizados para la clasificación.

4.5. Conclusiones y trabajo futuro

Se obtuvo una la plataforma web, gratuita y de código abierto, en su instancia de MVP, destinada al etiquetado de variantes y a un primer acercamiento al entrenamiento de usuarios en la tarea de clasificación de variantes. Dentro de los casos de uso planteados, los dos mencionados anteriormente son los que más se han desarrollado, cumpliendo con los objetivos mínimos establecidos para el proyecto. En este sentido, la plataforma alcanzó un estado preliminar, en el cual está pronta para comenzar a ser evaluada por usuarios. De esta forma, más allá de las evaluaciones preliminares realizas hasta ahora, se podrá evaluar el conjunto total de puntos débiles, fortalezas, usabilidad y viabilidad. Como abordaje futuro, se espera comenzar su evaluación y generación de datos en distintos grupos de trabajo, con diversidad de niveles de usuarios. El abordaje de la evaluación deberá realizarse desde diversos puntos de vista: evaluación de la interfaz, de la operatividad, del avance de los usuarios en niveles, de la estrategia de aprendizaje y su evolución, de las generaciones de consenso y su real aplicación, etc.
Desde el punto de vista de su operatividad, el sistema logra cumplir con los requerimientos planteados referidos a su integrabilidad, seguridad, y usabilidad. Uno de los requerimientos funcionales que requiere más trabajo a futuro es el de la internacionalización. La plataforma lograda pasó por distintos estadios en este sentido, iniciándose con una interfaz completamente en inglés, y luego siendo traducida a español. De todas formas, dado que la anotación de los datos es adquirida de distintas fuentes que los integran en inglés, el pasaje de idioma no logró completarse de forma satisfactoria. Se espera a futuro poder contar con una implementación que logre adaptar, al menos los aspectos principales, en español, a través de Flask o inicialmente a través de distintas páginas en Dash. Sería interesante poder sumar portugués como idioma posterior al español, contemplando potenciales colaboraciones en la región más cercana. Si bien en la versión preliminar la plataforma que se ha desarrollado hasta el momento no se cuenta con la habilitación de usuarios invitados y de registros nuevos en la plataforma, las vías para permitir la participación de los actores planteados en la especificación del proyecto se encuentran implementadas.
Los datos con los que cuenta la plataforma para una funcionalidad mínima pudieron adquirirse a partir de repositorios públicos, y lograron ser procesados con herramientas utilizadas habitualmente en la práctica de la genómica médica. Se estima que a futuro será necesario recurrir a otras estrategias, no solo para el procesamiento y anotación de los datos, sino como también para su almacenamiento. Las herramientas utilizadas para obtener la información destinada a los usuarios para que éstos puedan evaluar una variante, en su versión libre, cuentan con versiones desactualizadas de las bases de datos, y contar con versiones que cuenten con distintos niveles de actualización de los distintos datos requeridos pone en peligro la correcta realización de la tarea de clasificación. Si la información de las variantes es incorrecta, esto derivará en evaluaciones incorrectas, y por ende los algoritmos

que requerirán de estos datos brindarán predicciones incorrectas también (además de que el aprendizaje desde el usuario se puede ver afectado también). En este sentido, el abordaje futuro pretende independizarse de dicha anotación, tratando de abordar la integración de datos a través de distintas APIs y documentos. La idea es que el esquema de base de datos a futuro pueda acompañar este abordaje. Si bien dada la anotación fue directo abordar la base de datos con un esquema relacional, a futuro deberá tender a estrategias más flexibles. Se logró generar un conjunto de datos para una funcionalidad mínima, sin embargo el conjunto de variantes deberá ampliarse, incluyendo bases de datos de distintos orígenes, que puedan brindar variantes, sobre todo a la instancia de aprendizaje, pudiendo ser bases de datos de enfermedades (Fibrosis Quística, Cáncer, entre otras) a poblacionales. Tanto para el etiquetado de variantes como para su clasificación, la estrategia elegida de selección de variantes individuales se considera como preliminar, dada la importancia que tiene la evaluación de variantes en el contexto de un caso de estudio, ya sea un paciente o una familia. Es por ello que se espera que la plataforma pueda ser adaptada a la evaluación de casos, usándolos en distintas instancias de aprendizaje. Estos casos pueden en un principio obtenerse de los estudios locales realizados dentro del grupo de desarrollo, y se puede tender a la simulación de casos para ampliar las posibilidades de evaluación.

Desde el punto de vista de los casos de uso implementados, el servicio de clasificación de nuevas variantes, es el que se encuentra menos explorado y desarrollado, y que se espera que cobre protagonismo una vez que los insumos sean más amplios desde los usuarios. Si bien las bases para la aplicación de los algoritmos de predicción que se vayan refinando ya se encuentran, se requiere ampliar el alcance de la plataforma como comunidad. Las variantes que los usuarios ingresan se cargan en la plataforma, sin embargo los espacios para identificar el proceso que van siguiendo dichas variantes actualmente no se encuentran, y no hay una estructura que permita, por ejemplo, el intercambio de grupos o paneles de usuarios para trabajar en conjunto en la evaluación de una variante o una muestra. Esto refiere a la posibilidad de ampliar la plataforma como un servicio colaborativo real a distintas variantes, incluidas las de pacientes. En este sentido, como fue discutido, será importante definir si la plataforma se ampliará para dicha evaluación, en ese caso deberá incorporar más características como plataforma de interpretación.

La instancia de etiquetado logró completarse en sus requerimientos mínimos, cumpliendo con formas que el usuario evalúe una variante, brinde una etiqueta, la misma pueda ser almacenada y posteriormente procesada. No obstante, la generación de consenso alcanzó una implementación que no utiliza herramientas o estrategias del estado del arte, y que no reflejan la complejidad de la evaluación. En este sentido, se tenderá al desarrollo de herramientas que, como fue mencionado anteriormente, contemplen el nivel de pericia de los usuarios, el origen de las diferencias en la clasificación, las posibilidades de re-evaluación ante la evolución temporal de la información disponible, nivel de acuerdo inter e intra evaluadores, las distintas fuentes de ruido que pueden perjudicar la veracidad de las etiquetas proporcionadas, etc. Para comenzar a realizar dichos abordajes es necesario generar una base de conjuntos de datos para su análisis preliminar, por lo que puede hacerse en simultáneo a las evaluaciones futuras de la plataforma.

En lo que respecta a la instancia de aprendizaje, se logró la implementación de un primer abordaje, que implica, desde la selección de la información acorde y su organización para que el usuario pueda evaluar una variante correctamente, hasta algunos elementos simples de gamificación, como niveles de usuarios, sistema de puntos e interacción mínima. El abordaje, como fue mencionado anteriormente también fue inicial, pero permitió generar una base sobre la cual construir una estrategia más amplia y formal. Un elemento crucial a desarrollar en versiones posteriores es la interacción con el usuario, el desarrollo de tutoriales, guías, ayudas visuales, *feedback* más activo y continuo, y un mayor enfoque en el uso de las reglas a lo largo de toda la evaluación. Actualmente, un usuario que ya conozca las reglas de clasificación y se desempeñe en la tarea puede participar sin problemas, pero no ocurre lo mismo con alguien que esté comenzando a integrarse en el tema. Además de refinar los aspectos de gamificación ya implementados, como el avance a lo largo de los niveles a través de un sistema de puntos más adaptado a la tarea que se está realizando, se espera que se incorporen más elementos

como recompensas, desafíos, entre otros.
En una fase de prototipo, la plataforma cumple con los requisitos mínimos desde un punto de vista tecnológico, sin embargo, dadas las limitaciones expuestas anteriormente, en un posible escenario futuro de escalabilidad, se requerirá que se adopte otra estrategia para la implementación de las distintas partes de la plataforma. Desde la interfaz web, Dash cumplió con su propuesta de valor de ocultar la complejidad de escribir una aplicación web mediante lenguajes estándar, sin embargo resulta una herramienta limitada para el uso que se le quiere dar en este caso. Esto no es solo por la falta de control que se tiene sobre determinados aspectos de la aplicación generada, sino también por la integración del *dashboard* con una aplicación de mayores dimensiones. Dash es una herramienta para crear *dashboards*, y en el proyecto actual se requerían más funcionalidades, por lo que no corresponde, en este caso, atribuir a Dash las limitaciones identificadas, sino al proceso de selección de tecnologías. Si bien se considera que el aporte de Dash en cuanto a la disminución de tiempo de desarrollo no fue considerablemente grande, la implementación de la plataforma fue una buena oportunidad de la exploración y explotación de la herramienta, y la experiencia generada servirá para ser derivada a otros proyectos más adaptados. En este sentido, a futuro se podría tratar de implementar una aplicación clásica de React o mediante el uso de PHP. Más allá de los mencionado anteriormente, el planteo inicial consistía en que la interfaz alcance una instancia de PoC, lo cual fue satisfecho. Desde el *backend*, el enfoque cumple con generar una base robusta para servir a una interfaz de prueba de concepto, sin embargo se han observado limitaciones que fomentan la necesidad de probar otros enfoques, principalmente desde el punto de vista de la base de datos. Las características actuales de los datos y su procesamiento acompañan la decisión del trabajo en un esquema relacional, no obstante, una exploración a futuro interesante sería la adaptación de la base de datos a un esquema no relacional o basado en documentos, que se sirva de APIs. Por otro lado, la API cuenta con una implementación robusta, que hoy en día permite que el sistema pueda ser usado por terceros que accedan directamente a los *endpoints* de la misma.
Como conclusión final, en base a los aspectos mencionados anteriormente se puede mencionar que se logró alcanzar una primera implementación de un producto final que satisface el alcance mínimo establecido inicialmente. En las evaluaciones realizadas se pudo valorar que las herramientas elegidas y las estrategias pueden ser mejoradas a los efectos de alcanzar el estado del arte. No obstante, se considera que más allá de que el producto final no sea actualmente competente en el mercado y tenga varios aspectos en los que trabajar, el proyecto permitió obtener dimensión le lo que se requiere para implementar desde cero un sistema con las características planteadas al inicio, y una serie de conocimientos muy valiosos para dicho objetivo. Se considera que el desarrollo del trabajo permitió brindar un acercamiento importante y manipulación de herramientas y datos que son comunes en el desarrollo de herramientas web y proyectos a grande escala: herramientas de desarrollo de interfaces web, de procesamiento, análisis y visualización de datos, herramientas para el desarrollo de APIs, base de datos y de gestión y control de versiones de archivos. Además, el hecho de generar una herramienta para intentar transmitir el proceso de caracterización de una variante y/o proceso diagnóstico en general a usuarios en distintos niveles, requirió generar y reforzar una base relativamente amplia de conocimientos en dichos procesos y en la base biológica subyacente. El trabajo en el proyecto no solo permitió dimensionar los requerimientos para su implementación desde un punto de vista de tecnologías aptas para ello y los datos necesarios, sino también desde una perspectiva de recursos humanos. Más allá de la cantidad de recursos mediante los cuales se puede llegar a una implementación completa y viable, se destaca la necesidad de la transdisciplina desde la concepción del proyecto, y cómo su abordaje requiere de la participación activa desde quienes se desempeñan en el desarrollo de las herramientas, y el cliente aportando su visión clínica.

Capítulo 5

Clasificación de variantes

5.1. Introducción

Las técnicas de aprendizaje automático se han utilizado ampliamente en la investigación genómica [113]. Las mimas se dividen en dos grandes categorías principales: supervisadas y no supervisadas. En el aprendizaje supervisado, el objetivo es predecir la etiqueta (clasificación) o la respuesta (regresión) de cada punto de datos utilizando un conjunto de ejemplos de entrenamiento etiquetados [243]. Los métodos supervisados requieren datos con etiquetas observadas (por ejemplo, etiqueta patogénica o benigna de una variante) que pueden utilizarse para predecir etiquetas no observadas para nuevos datos. Por otro lado, los métodos de aprendizaje no supervisado, extraen patrones de las características de los datos y no necesitan etiquetas. En este sentido, en métodos como *clustering* o el análisis de componentes principales, el objetivo es aprender patrones inherentes a los propios datos.

La segunda parte del presente trabajo consistió en la exploración de algoritmos clásicos de aprendizaje automático supervizado aplicados a la clasificación de variantes cortas en la línea germinal, con énfasis en variantes en regiones codificantes del genoma. Esta exploración resultaría un puntapié inicial para posteriores análisis de nuevas estrategias de clasificación, y su uso conjunto con la plataforma planteada anteriormente. El objetivo final consiste en contar con un sistema completo con datos provenientes de la plataforma a partir de usuarios con un entrenamiento y niveles acordes y bien calculados, y la generación de consenso que permita contemplar el peso de los usuarios como insumo. En lo que compete a la instancia de este trabajo, la evaluación inicial no contemplará datos de la plataforma, ni una generación de consenso final (dado el estado preliminar de dichos aspectos), sino que consistirá en la evaluación de métodos a partir de datos provenientes exclusivamente de bases de datos públicas, con etiquetas generadas y obtenidas a partir de las mismas.

5.1.1. Objetivos

5.1.1.1. Objetivo general

Implementar y evaluar un mecanismo de clasificación de variantes cortas en el genoma, provenientes de la línea germinal, basado en métodos de aprendizaje automático.

5.1.1.2. Objetivos específicos

- Obtener y definir un conjunto de datos adecuado de variantes para entrenar y evaluar los modelos implementados/evaluados.
- Lograr una caracterización de los tipos de datos trabajados, identificando cuáles son sus limitaciones al momento de implementar métodos de aprendizaje automático.
- Buscar estrategias para atacar las limitaciones identificadas (teniendo en cuenta el punto anterior).

- Explorar distintos algoritmos de aprendizaje automático aplicado al conjunto de datos definido y procesado, evaluando la precisión de cada uno.
- Seleccionar un método de clasificación entre los explorados para ser aplicado a la priorización, en un principio, de variantes cortas en la línea germinal.

5.2. Metodología

Como fue explicado anteriormente, el funcionamiento esperado de un método de clasificación final es en tiempo real, tomando datos de entrenamiento a partir de la base de datos descrita anteriormente. Teniendo en cuenta el desarrollo en simultáneo de las estrategias de clasificación y de la plataforma, la metodología presentada a continuación consiste en el uso de un sub-conjunto de datos de dicha base de datos, correspondientes a una versión previa de la misma.

5.2.1. Pre-procesamiento de datos

Previamente a la aplicación de diferentes técnicas de aprendizaje supervisado sobre los datos, fue necesario dejar los datos aptos para dichos procesos.

5.2.1.1. Descripción y preparación de datos

Los datos principales utilizados en esta instancia del proyecto corresponden a ClinVar (mayo de 2021). Inicialmente se contó con un total de 780166 variantes, seleccionadas y divididas según su significado clínico, como se muestra en la Tabla 5.1 . Preliminarmente se buscará evaluar el comportamiento de los distintos algoritmos únicamente en variantes exónicas o que se encuentran en regiones codificantes del genoma. Para ello, las variantes exónicas fueron seleccionadas inicialmente a partir de determinadas columnas de anotación (a partir de Refseq). Una vez que se contó con un conjunto de variantes exónicas, se obtuvieron las seis clases mostradas en la Tabla 5.1 según el significado clínico. Para la selección de variantes según su significado clínico no se tomaron en cuenta combinaciones de etiquetas, a excepción de las variantes de significado conflictivo, en las que se agruparon los conflictos incluso en combinaciones. Cabe destacar que si bien en la plataforma se tenía en cuenta la posibilidad brindar más etiquetas que las que fueron seleccionadas en este caso, el caso de la plataforma se amplió a posibilidades de discriminar entre más clases a futuro. En el caso de este capítulo, el enfoque está en la discriminación entre las 5 clases inicialmente sugeridas por ACMG/AMP [44].

Tipo	Codificantes	No codificantes	Total
Benigna	52527	36774	**89301**
Probablemente Benigna	145877	43896	**189773**
Patogénica	43492	8177	**51669**
Probablemente Patogénica	22626	8196	**30822**
VUS	310175	63618	**373793**
Interpretación conflictiva	37697	7111	**44808**
Total	**612394**	**167772**	**780166**

Cuadro 5.1: Distribución de tipos de variantes según su significado clínico o etiqueta de patogenicidad, luego de filtrados aquellos tipos de interés para el proyecto actual. Si bien en esta instancia se trabajará únicamente en la clasificación de variantes codificantes, se presenta un resumen de las variantes no codificantes a los efectos de mostrar los números totales en comparación.

El conjunto de datos contó inicialmente con aproximadamente 200 columnas, correspondientes a la anotación general presentada en el Capítulo 3, a partir de la herramienta ANNOVAR. En relación a la descripción de la base de datos realizada en el Capítulo 4, los datos con los que se cuenta para entrenar y evaluar los modelos corresponden a la tabla de variantes ya mencionada. Parte de la preparación de los datos implica la selección de las características que aporten información relevante a la clasificación y aquellas cuyos valores categóricos puedan ser codificados manteniendo el sentido biológico de la información que contienen. Entre las columnas eliminadas se encuentran aquellos predictores específicos de variantes no codificantes que no cuentan con valores en variantes codificantes. En este sentido, se eliminaron aproximadamente 135 columnas de la tabla inicial, habiendo seleccionado las características iniciales presentadas en la Tabla 5.2 para el entrenamiento y evaluación de los modelos.

Las características usadas se pueden agrupar de acuerdo a la categoría o la naturaleza de los datos, tal como se muestra en la Tabla 5.2. Por un lado se cuenta con información a nivel poblacional, la cual está dada principalmente por frecuencias alélicas adquiridas de varios proyectos. La frecuencia alélica menor (MAF) de estas bases de datos suele ser un indicador muy útil en priorización, y una característica que se ha reportada como de relevancia para la creación de modelos de predicción [244]. En esta instancia se filtraron de la anotación general las frecuencia correspondientes a las sub-poblaciones presentes para cada proyecto, seleccionando las frecuencias totales y máximas para 1000 Genomas, ExAC, gnomAD, ESP y Kaviar. Además se consideró la frecuencia máxima calculada a partir todas las frecuencias máximas de cada proyecto. Además de la información poblacional, se incluyeron *scores in sílico* de predicción de patogenicidad. Estos *scores*, como ya fue descrito anteriormente en el Capítulo 4, a su vez se agrupan en herramientas de predicción a través del impacto físico-químico de la variante, y predictores de la conservación evolutiva. Entre estos predictores, se consideraron SIFT, Polyphen2, LRT, PROVEAN, CADD, DANN, MutationTaster, MutationAssesor, FATHMM, FATHMM-MKL, VEST3, MetaSVM, MetaLR, M-CAP, REVEL, Eigen, fitCons, PhyloP, PhastCons y GERP. Además, se incluyen características referidas al impacto funcional de la variante, y aquellas relacionadas con las reglas ACMG/AMP.

Entre las características presentadas, dos se desprenden de la anotación, pero fueron calculadas explícitamente para su uso en esta instancia: el *Grantham Score* y *AGMG score*. El cálculo del *Grantham Score* se hace sobre cada variante no sinónima, a partir de la anotación en modo *gene-based-annotation*, descrita en el Capítulo 3, añadiendo en ANNOVAR el argumento `--aamatrixfile`, ingresando en este caso una matriz de sustitución Grantham [245]. La tabla final permite extraer el *score*, y finalmente se une esta información a la tabla de anotación original. En el caso del *AGMG score*, el mismo se desprende de la anotación proveniente de Intervar, que aporta valores booleanos en función del cumplimiento de las 28 reglas ACMG/AMP. Para su implementación el presente trabajo se basó en [4], en donde se desarrolla un *score* que resulta de la suma ponderada de la evidencia de las reglas. Los pesos usados para dicha suma corresponden a `BA=-9, PVS=6, PS=4, PM=2, PP=1, BS=-3, BS2=-3, BP3=-1, BP4=-1, BP7=-2`. El principio para la selección de los pesos deriva de las reglas ACMG/AMP, específicamente de la recombinación de criterios [44] [4] (ver Apéndice A).

Categoría	Características	Definición	Tipo
Poblacional	`freq_max`	Frecuencia máxima poblacional calculada a partir de todas las frecuencias disponibles en todas las poblaciones, para todos los proyectos para los que se cuenta con información de frecuencia poblacional.	*Float*

Tabla 5.2 continúa de la página previa

Categoría	Características	Definición	Tipo
	esp6500siv2_all	Frecuencia poblacional para todas las poblaciones en el proyecto ESP.	*Float*
	1000g2015aug_all	Frecuencia poblacional para todas las poblaciones en el proyecto 1000 Genomas [73].	*Float*
	ExAC_ALL	Frecuencia poblacional para todas las poblaciones en el proyecto ExAC [76].	*Float*
	gnomAD_genome_AF_popmax	Frecuencia máxima poblacional para todas las poblaciones en el proyecto gnomAD [107].	*Float*
	Kaviar_AF	Frecuencia alélica para las variantes de todas las poblaciones en el proyecto Kaviar	*Float*
Predictores in silico	grantham_score	Predicción de la distancia entre dos aminoácidos, en un sentido evolutivo [245].	Float
	SIFT_score	Predicción del efecto de la sustitución en la función de la proteína [246].	*Float*
	Polyphen2_HDIV_score	Predicción de efecto deletéreo de una variante según HumDiv [124].	*Float*
	Polyphen2_HVAR_score	Predicción de efecto deletéreo de una variante según HumVar [124].	*Float*
	LRT_score	Predicción funcional para variantes no sinónimas de LRT proporcionada por dbNSFP [247].	*Float*
	MutationTaster_score	Predicción funcional para variantes no sinónimas [248]	*Float*
	MutationAssessor_score	Predicción del impacto funcional de sustitución de aminoácidos [249]	*Float*
	FATHMM_score	Predicción de consecuencias funcionales de variantes codificantes y no codificantes [133]	*Float*
	PROVEAN_score	Predicción del impacto de una sustitución o indel en la función biológica de una proteína [250].	*Float*
	VEST3_score	Predicción de la importancia funcional de las mutaciones *missense* en función de la probabilidad de que sean patogénicas [251].	*Float*
	MetaSVM_score	Predicción de impacto deletéreo en variantes basado en regresión logística [7]	*Float*
	MetaLR_score	Predicción de impacto deletéreo en variantes basado en regresión logística [7]	*Float*

Tabla 5.2 continúa de la página previa

Categoría	Características	Definición	Tipo
	M-CAP_score	Predicción de patogenicidad de variantes raras *missense* [127]	*Float*
	REVEL_score	Predicción de patogenicidad de variantes *missense* basado en una combinación de 13 herramientas individuales [128]	*Float*
	CADD	Score de predicción de patogenicidad en variantes cortas [132].	*Float*
	DANN	Predicción de patogenicidad en variantes basado en DL [125]	*Float*
	fathmm-MKL_coding_score	Predicción de consecuencias funcionales de variantes codificantes y no codificantes [126]	*Float*
	Eigen	Scores de integración de anotaciones funcionales para variantes codificantes y no codificantes [158].	*Float*
	integrated_fitCons_score	Medida evolutiva de la función genómica potencial [252]	*Float*
	GERP++_RS	Medida de conservación de la secuencia [253]	*Float*
	phyloP100way_vertebrate	Medida de conservación a partir de alineamiento de secuencias de 100 especies de vertebrados [254].	*Float*
	phyloP20way_mammalian	Medida de conservación evolutiva a partir de alineamiento de secuencias de 20 especies de mamíferos [254].	*Float*
	phastCons100way_vertebrate	Medida de conservación a partir de alineamiento de secuencias de 100 especies de vertebrados, basada en HMM [255].	*Float*
	phastCons20way_mammalian	Medida de conservación evolutiva a partir de alineamiento de secuencias de 20 especies de mamíferos, basada enHMM [255].	*Float*
	SiPhy_29way_logOdds	Medida de conservación evolutiva a partir de genomas de 29 mamíferos [256]	*Float*
Funcional	ExonicFunc.refGene	Consecuencia de la variante [148].	*String*
Reglas ACMG	PVS1, PS1-4, PM1-6, PP1-5, BA1, BS1-4, BP1-7	Evaluación de cada regla ACMG/AMP [44]	*Booleano*
	ACMG_score	Suma ponderada de la evidencia aportada por cada regla ACMG/AMP.	*Entero*

Tabla 5.2 continúa de la página previa

Categoría	Características	Definición	Tipo

CUADRO 5.2: Características consideradas inicialmente para el entrenamiento de modelos. Las características se dividen, al igual que en la instancia de la plataforma de clasificación, según la naturaleza de los datos en: Información poblacional, *scores* de predicción, efecto funcional de las variantes en el genoma y evaluación según las reglas ACMG.

Encoding

Anteriormente se mencionó que se filtraron algunas características categóricas cuyos valores no podían ser codificados a valores numéricos. En este sentido se mantuvo de las características de tipo categórico y fue posteriormente pasada a valores numéricos, a los efectos de mantenerla. La variable a transformas de datos categóricos a numéricos correspondió a `ExonicFunc.refGene`, la cual representa la consecuencia funcional de la variante. Para poder representar el peso de cada categoría reflejando un punto de vista biológico, se sustituyó cada una de las categorías presentes con un valor que represente el impacto, donde 1 representa el mayor impacto a nivel genético, y 0 representa una mutación con un impacto imperceptible a nivel genético o del producto funcional. En la Tabla 5.3 se presenta cómo se codificaron los valores de la variable. La generación del *encoding* se hizo de forma arbitraria, habiendo probado distintas combinaciones de valores asignados a cada categoría. Si bien los resultados presentados corresponderán a los valores presentados en la Tabla 5.3, se consideró además la codificación alternativa de *stoploss* con un valor inferior a 1 (0.6), teniendo en cuenta un posible impacto menor respecto a un cambio que produzca una ganancia de codón *stop*.

Valor original	Código
`synonymous_SNV`	0
`nonsynonymous_SNV`	0.33
`nonframeshift_deletion`	0.66
`nonframeshift_insertion`	0.66
`nonframeshift_substitution`	0.66
`frameshift_deletion`	1
`frameshift_insertion`	1
`frameshift_substitution`	1
`stopgain`	1
`stoploss`	1

CUADRO 5.3: Transformación de los valores categóricos de efecto de las variantes (`ExonicFunction.refGene`) a valores numéricos. Los valores se encuentran en un rango entre 0 y 1, donde 0 representa un efecto no deletéreo, y 1 representa los impactos mayores.

5.2.2. Manipulación de valores faltantes

Una vez que se contó con un conjunto de datos inicial, se procedió a la evaluación de variantes y columnas con valores faltantes. Como es parte de la naturaleza de los datos biológicos, la presencia de valores faltantes tiene un gran impacto en conjuntos de datos genómicos. En lo que respecta a este trabajo en particular, dado que la anotación depende de que los valores puedan obtenerse de otras bases de datos, dependiendo de la naturaleza de las variantes es altamente probable que hayan datos incompletos, o que no hayan sido reportados o identificados.

Una vez que se detecta la fuente de los valores faltantes, hay distintos tipos de herramientas o estrategias que pueden ser utilizadas para atacar este problema, las cuales serán descritas a continuación [257].

Supresión de valores

Esta estrategia implica eliminar los valores que faltan en un conjunto de datos. En los dos primeros casos descritos anteriormente, se considera seguro eliminar los datos con valores faltantes en función de su frecuencia, mientras que en el tercer caso, si los datos que faltan no son aleatorios, eliminar las observaciones con valores faltantes puede producir pérdidas de información, derivando a sesgos en el modelo. Las el borrado de datos con valores faltantes puede ser de tres tipos:

- Eliminación por listas (*listwise*): Este enfoque consiste en eliminar cualquier muestra o fila (teniendo en cuenta una estructura de tabla) con uno o más valores omitidos, lo que da lugar a un conjunto de datos más pequeño.
- Eliminación por pares (*pairwise*): Este enfoque elimina sólo los valores que faltan para cada característica, lo que da como resultado un conjunto de datos con algunas muestras que tienen valores que faltan. Puede preservar más información que la eliminación por listas, pero también puede dar lugar a resultados sesgados.
- Eliminación de columnas: Si una columna contiene una proporción elevada de valores faltantes, y la característica no es significativa, se puede considerar eliminar dicha característica.

En general, más allá de las características de los datos, es común optar por estrategias de imputación, preferentemente por sobre la eliminación de datos o características.

Imputación

La imputación consiste en sustituir los datos que faltan por valores generados. Hay muchas formas de imputar los valores faltantes en función de la naturaleza del problema y de los datos. Dependiendo de la naturaleza del problema, las técnicas de imputación pueden clasificarse según: uso general o básico, para problemas de series temporales, o uso avanzado.

- Estrategias básicas: las estrategias básicas de imputación implican el uso de valores constantes o de estadísticos (media global, la mediana o la moda de la característica) para completar los valores que faltan. Si bien estas estrategias son rápidas de aplicar y muy utilizadas, en algunos casos puede reducir la varianza de los datos, además de que no es menor que en la aplicación actual puede perder sentido desde un punto de vista biológico.
- Imputación para problemas de series temporales: la imputación de valores faltantes en el caso de problemas de series temporales tiene un enfoque distinto al seguido en otros casos, y sin entrar en detalles (dado que no es el enfoque de este trabajo), se pueden considerar tres tipos de estrategias, sustituir los valores faltantes con la última observación, sustituirlos con la siguiente observación, o métodos de interpolación lineal.
- Estrategias avanzadas: Las técnicas avanzadas de imputación utilizan algoritmos de aprendizaje automático para imputar los valores ausentes. Esto es a diferencia de las técnicas anteriores, en las que se utilizan los valores de otras columnas para predecir los valores faltantes. Entre estas estrategias se encuentran métodos como imputación mediante KNN, o imputación multivariada, conocida como MICE (*Multivariate Imputation by Chained Equations*). En el caso de imputación mediante KNN cada característica faltante se imputa usando los valores de los K-vecinos más cercanos que cuentan con un valor para la característica, y las características de

los vecinos se promedian uniformemente, o se ponderan por la distancia a cada vecino [257]. En el caso de MICE, permite imputar valores faltantes modelando cada característica con valores perdidos en función de otras características en forma rotatoria. Realiza regresiones múltiples sobre una muestra aleatoria de los datos y, a continuación, toma la media de los valores de regresión múltiple y utiliza ese valor para imputar el valor que falta.

Uso de modelos que pueden gestionar valores faltantes

Por último, algunos algoritmos, pueden tratar los valores que faltan sin necesidad de pre-procesamiento, suministrando los parámetros pertinentes. En esta categoría entran algoritmos como como XGBoost [258] y LightGBM [259]. En el caso de XGBoost, la potencialidad de su buen desempeño con valores faltantes se basa en su característica como modelo *Sparsity-aware split finding*, mediante la cual las direcciones de las ramas para los valores que faltan se aprenden durante el entrenamiento.

5.2.2.0.1. Estrategias a aplicar

En el presente trabajo fueron exploradas algunas de las estrategias mencionadas anteriormente [257]. El flujo de trabajo con estos datos implica la evaluación del escenario de datos faltantes en el problema actual y la evaluación de estrategias a aplicar. Al momento de aplicar estrategias de manipulación de valores faltantes se consideró la eliminación de variantes con un elevado contenido de valores ausentes (se eliminaron datos con menos de un 70 % de sus características anotadas), la eliminación de columnas con proporciones elevadas de valores faltantes ($> 80\,\%$), además de considerar la imputación de los valores faltantes restantes. Para la imputación, se consideró un trabajo realizado previamente en conjuntos de datos similares, en los que se evaluaron metodologías de imputación de valores faltantes en variantes no codificantes [260]. En este tipo de datos, el gran número de valores o anotaciones faltantes requirió de un flujo de trabajo riguroso y un énfasis importante en la imputación. Si bien las variantes codificantes manipuladas en el presente trabajo en general contienen una proporción menor de valores ausentes respecto a las no codificantes (dada la compleja naturaleza de estas últimas), se comparten características en los datos en general, por lo que se utilizarán las estrategias que mejor se adaptaron en este caso. En el trabajo realizado en [260] se identificó que los mejores resultados estaban dados por los algoritmos *Random Forest* y KNN. Tomando esto como base, se implementó la imputación de valores faltantes de las columnas usando KNN con $k = 10$.

5.2.3. Análisis de características

Una vez que se obtuvo un conjunto de datos sin valores faltantes se procedió a estudiar la correlación de variables. Para ello se utiliza el coeficiente de correlación estándar (también llamado r de *Pearson*) entre cada par de atributos utilizando el método `corr()` en Python. En el presente trabajo se considerará como variables correlacionadas como aquellas cuyos pares tengan más de un 90 % de correlación según *Pearson*. En caso de que se identifique una correlación alta entre las variables, se puede decir que las mismas tienen el mismo efecto en los modelos de clasificación podría sobrestimar la importancia de determinadas característicos. Si una misma característica biológica está representada por varias columnas, entonces los métodos estarán sesgados de acuerdo a la variación en dicho conjunto de características.

5.2.4. Conjuntos de entrenamiento, validación y *test*

Previo a la separación en conjuntos de entrenamiento, validación y test se manipuló el conjunto de datos totales a los efectos de balancear las clases. Para esto se eliminaron datos de forma aleatoria de los conjuntos sobrerrepresentados. Si bien se evaluaron técnicas de aumentación de datos para las

clases minoritarias (específicamente SMOTE[1]), en el presente trabajo se presentarán los resultados sin técnicas de aumentación.

Para generar los distintos conjuntos de datos se exploraron diferentes estrategias se separación de los mismos, y a los efectos de este trabajo, se presentarán dos de ellas. La primera estrategia evaluada fue la separación aleatoria de los distintos conjuntos de datos. A su vez, para esta separación aleatoria se plantearon distintos métodos y proporciones. En este sentido, se evaluaron métodos como `train_test_split` de *Sklearn* o `train_valid_test_split` de *Fast_ml* y la selección aleatoria simple. Finalmente se optó por trabajar con una selección aleatoria simple, a los efectos de manipular el balance de clases en cada conjunto de datos. Las proporciones seleccionadas fueron: 75 % para el conjunto de entrenamiento, 15 % para el conjunto de validación y 10 % para el conjunto de prueba o *test* . La segunda estrategia evaluada para la separación de datos implica la contemplación de las fechas de publicación de las variantes, teniendo en cuenta distintas versiones de ClinVar y de la anotación. Esto se realizó con el fin de evitar la contaminación de información y el uso de la misma tanto en el conjunto de evaluación como en los conjuntos de validación y *test* (a los efectos de prevenir el sobreajuste de los modelos).
Para cada modelo a su vez se evaluaron distintos escenarios de agrupación de los datos, en los que difirió el conjunto de entrenamiento y evaluación utilizado. Uno de los escenarios implicó la clasificación binaria entre variantes patogénicas y no patogénicas. En este caso se evaluaron los distintos métodos únicamente con las clases "puras" y luego con combinación de clases en las que se unieron probablemente benignas con benignas y probablemente patogénicas con patogénicas. Otro de los escenarios implica el trabajo en clasificación multi-clase, a través de la evaluación de la patogenicidad en las 5 clases establecidas para ACMG/AMP: Benigna (B), Probablemente Benigna (PB), Patogénica (P), Probablemente Patogénica (PP) y VUS.

5.2.5. Entrenamiento y selección de modelos

Una vez que se contó con el conjunto de datos a trabaja para cada instancia y tipo de prueba, se pasaron a evaluar distintos algoritmos de aprendizaje supervisado. Las pruebas realizadas sobre el conjunto de datos se hicieron desde la exploración de los modelos más simples (como regresión), hasta acercamientos preliminares al uso de redes neuronales. A través de la evaluación, se pasaron por distintas instancias no solo de complejidad de los modelos, sino también de posibilidades de interpretabilidad. Más allá de todas las evaluaciones realizadas, en el presente documento se presentarán solo algunos de los modelos evaluados, en particular los que presentaron un mejor desempeño y aplicabilidad. En particular se considerarán aquellos modelos que ya han sido explorados en el área, a los efectos de evaluar su desempeño en el conjunto de datos con el que se cuenta en este trabajo.

Los métodos explorados a presentarse en este trabajo son modelos de clasificación de aprendizaje supervisado: regresión logística, K-Vecinos más cercanos (KNN), *Naive Bayes*, *Support Vector Machines* (SVM), Árboles de decisión, *Random Forest*, y *Gradient Boosting Trees*. Cada modelo se exploró inicialmente con sus parámetros por defecto para cada estrategia de conjunto de dato evaluada. Cada modelo fue evaluado inicialmente con sus parámetros por defecto, para luego realizar el ajuste de hiperparámetros correspondiente:

- KNN: se utilizó mediante `sklearn.neighbors.KNeighborsClassifier`, con sus parámetros por defecto *n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric = 'minkowski'* [261]
- *Naive Bayes*: se utilizó mediante `sklearn.naive_bayes.GaussianNB`, con sus parámetros por defecto *priors=None, var_smoothing=1e-09* [262].

[1] *Synthetic Minority Oversampling Technique*

- SVM: se utilizó mediante `sklearn.svm.SVC`, con sus parámetros por defecto *C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0*, entre otros [263].
- *Decision tree*: se utilizó mediante `sklearn.tree.DecisionTreeClassifier` con sus parámetros por defecto *criterion='gini', splitter='best'*, entre otros [264].
- *Random Forest*: se utilizó mediante *sklearn.ensemble.RandomForestClassifier*, con sus parámetros por defecto *n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None, min_impurity_decrease=0.0*, entre otros [265].
- *GBDT*: mediante *xgboost.XGBClassifier*, con sus parámetros por defecto *objective = 'binary:logistic', use_label_encoder=None*[266].

Luego, se implementaron técnicas para selección de modelo y selección de los mejores hiperparámetros para cada conjunto de datos de estudio. Por un lado se aplicó la evaluación mediante validación cruzada (*crossvalidation*) para evaluar la generalización de los modelos a nuevos datos. El objetivo de la validación cruzada es estimar el rendimiento de un modelo con datos nuevos y desconocidos. Esto ayuda a reducir el sobreajuste y proporciona una estimación más precisa del rendimiento del modelo. Esta técnica consiste en dividir los datos en K conjuntos o *folds* de igual tamaño y entrenar el modelo K veces, utilizando cada vez un *fold* diferente como conjunto de validación y los K-1 *folds* restantes como conjunto de entrenamiento. El rendimiento del modelo se mide promediando los *scores* de los K conjuntos de validación[267]. En la Figura 5.1 se muestra una representación esquemática de dicho proceso. En el caso del presente trabajo se usan funciones pertenecientes al módulo `sklearn.model_selection` de *Scikit-learn*, para implementar una 10-*fold crossvalidation*, es decir, que el conjunto de entrenamiento se divide en 10 subconjuntos, se entrena y evalúan los modelos 10 veces, eligiendo un conjunto diferente para la evaluación acá vez y entrenando en el resto de los 9 conjuntos. A través de la validación cruzada no sólo se obtuvo una estimación del rendimiento de cada modelo, sino también una medida de su precisión (a través de la medida de la desviación estándar) [268]. En el caso del uso de validación cruzada, si bien debe seguir existiendo un conjunto de *test* para la evaluación final, el conjunto de validación ya no es necesario.

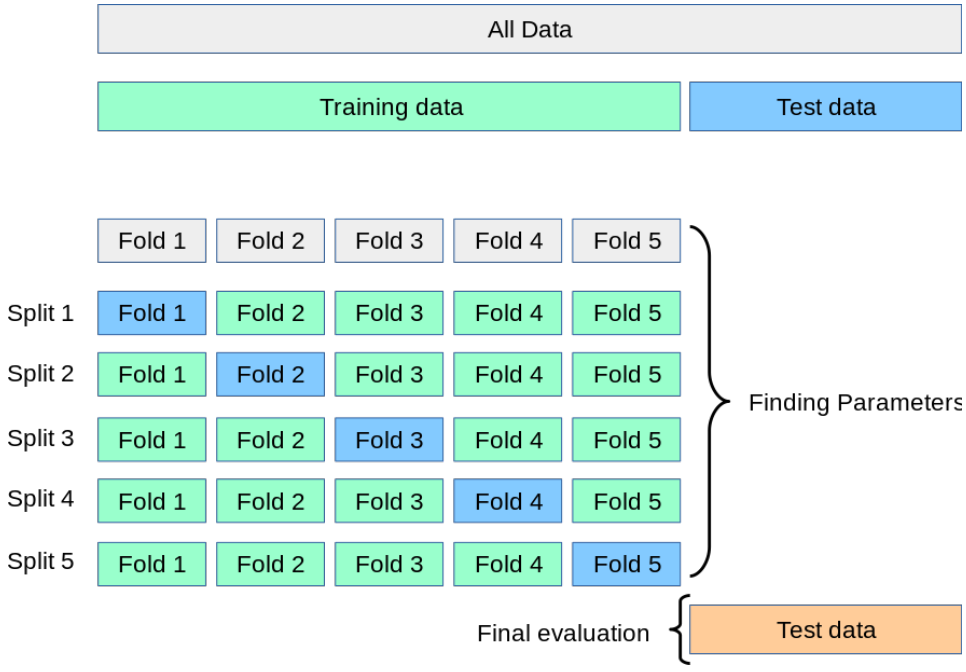


FIGURA 5.1: Diagrama del proceso de *k-fold crossvalidation*. Extraído de [268].

La validación cruzada nos permite comparar el rendimiento de distintos modelos y seleccionar el más adecuado para el problema. Una vez seleccionado el mejor modelo, se utiliza *Grid Search* para ajustar sus hiperparámetros y mejorar el rendimiento del modelo. *Grid Search* consiste en especificar un conjunto de hiperparámetros y sus posibles valores y, a continuación, buscar exhaustivamente entre todas las combinaciones posibles de estos valores para encontrar la combinación que dé como resultado el mejor rendimiento. En este caso, la técnica también se aplicó a través del módulo `sklearn.model_selection`. Los hiperparámetros suelen especificarse como un diccionario de nombres de parámetros y el rango de valores sobre el que se va a buscar [267] [269].

Después de ajustar los modelos, se obtiene un sistema que se considera como definitivo en esta instancia, y el mismo deberá ser evaluado en el conjunto de prueba (*test set*).

5.2.6. Implementación

El pre-procesamiento de los datos y la evaluación de los modelos se realizan en Python. Si bien ya fue mencionado anteriormente, la implementación de los modelos y herramientas de aprendizaje automático se hacen a través de *Scikit-learn* [270]. *Scikit-learn* también conocido como `sklearn`, es una librería de código abierto de Python para dedicada al aprendizaje automático. Ofrece una amplia gama de algoritmos y herramientas para tareas como clasificación, regresión, *clustering*, reducción dimensional, entre otras. Esta herramienta está construida sobre otras librerías populares de computación científica en Python, como NumPy, SciPy y Matplotlib, y está diseñada para integrarse perfectamente con estas librerías (las cuales también son usadas en el desarrollo del trabajo) [270]. *Scikit-learn* incluye una serie de algoritmos populares de aprendizaje automático, como los mencionados anteriormente y evaluados en este trabajo, entre muchos otros. Además, como ya fue visto, proporciona herramientas para la extracción de características, selección de características, pre-procesamiento de datos, evaluación de modelos y ajuste de parámetros.

5.2.7. Integración

En la Figura 4.3 del Capítulo 4 se presentó cómo interactuaría el modelo implementado con la plataforma web. Si bien actualmente no se cuenta con el ingreso de datos al algoritmo, en la instancia actual se puede solicitar una clasificación a través de la plataforma.
Una vez que se obtuvo el modelo final entrenado sobre todo el conjunto de datos, el mismo fue exportado a través de la librería `pickle` de Python. Luego el modelo se integra en el entorno de la plataforma *actg-Learn*, siendo usado para hacer predicciones a través de la web. Una vez que el usuario ingresa a la instancia de "Clasificación" de la plataforma, y carga nuevos datos, se envía una consulta a través de la API que llama al método `predict` de la librería. Además, cada vez que se generan nuevas versiones del modelo (en función de mejoras en el desempeño respecto al anterior), el mismo se actualiza.

5.3. Resultados y discusión

Los resultados presentados a continuación representan parte del análisis y exploración globales realizados, los cuales corresponden a los resultados más relevantes obtenidos en las últimas instancias.

5.3.1. Imputación de valores faltantes

En la exploración inicial de datos se buscó caracterizar los mismos, identificando los patrones de valores faltantes, para así identificar qué estrategia se seguiría. Del total de las variantes iniciales, menos del 2 % cuentan con el total de sus características cubiertas. Esto se debe a varios factores,

entre los que se identifican los patrones de anotación, el nivel de actualización en las bases de datos seleccionadas para la anotación, y las características intrínsecas de las variantes que hacen que determinados valores seleccionados no les correspondan. Por ejemplo, en la base de datos se cuenta con un variantes que generan la ganancia o pérdida de un codón *stop*. Ese tipo de variantes, por ejemplo, no contarán con valores para predictores que evalúen el efecto de sustituciones de aminoácidos (como PROVEAN). No obstante, esto no quiere decir que estas sean las únicas variantes para las que este valor no se encuentre. Como se puede observar en la Figura 5.2, hay varios casos en los que las proporciones de valores faltantes se comparten entre columnas, y esto se debe a que los patrones de valores faltantes muestran una fuerte correlación entre las mismas (Figura 5.3). Por ejemplo, en el dendrograma de la Figura 5.3 se puede ver como las hojas de uno de los *clusters* unidas a una distancia nula predicen completamente la presencia de la otra. En el caso de las reglas ACMG, tiene sentido que todas las variables estén completas siempre que una de ellas la esté, dado que refieren al mismo enfoque y provienen de la misma base de datos. Esto se repite en algunos predictores que provienen de la misma fuente de anotación, por lo tanto por ejemplo, las fechas pueden estar sincronizadas. En este sentido dado el mecanismo de obtención de los datos principalmente, no podemos decir que los datos tienen patrones de valores faltantes de forma aleatoria, por lo que se plantean estrategias de imputación para los mismos.
Los datos faltantes serán imputados, sin embargo aquellos datos que contengan proporciones elevadas de valores faltantes serán eliminados, a los efectos de que la imputación no tenga un impacto tal que se pierda el carácter biológico de los datos. El conjunto de datos final (luego de la eliminación de variantes que contengan más de un 30 % de valores faltantes) cuenta con las con un total de 389.447 variantes, distribuidas por clases según se muestra en la Tabla 5.4, y con un total de 59 características obtenidas tras eliminar aquellas que superan el 80 % de valores faltantes. Sobre estos datos se aplicó imputación de valores faltantes mediante KNN. Es importante destacar que los valores faltantes en determinadas columnas tiene una gran relevancia, como en las de frecuencias. En este caso, que los datos no se encuentren implican que se pueda asumir como cero, y esto cuenta con un peso primordial al momento de clasificar una variante (como fue explicado en función de las reglas). En este sentido, si bien se presentan las características de frecuencias incluidas en la caracterización de valores faltantes, los mismos fueron considerados como 0, a los efectos de evitar valores indeseados en la imputación.
El porcentaje de valores faltantes si bien es grande intra-categorías, es pequeño considerando que las regiones codificantes en el genoma (de donde provienen las variantes usadas) representan aproximadamente un 1 % del mismo, y que las variantes presentes en dichas regiones son las más identificadas y caracterizadas.

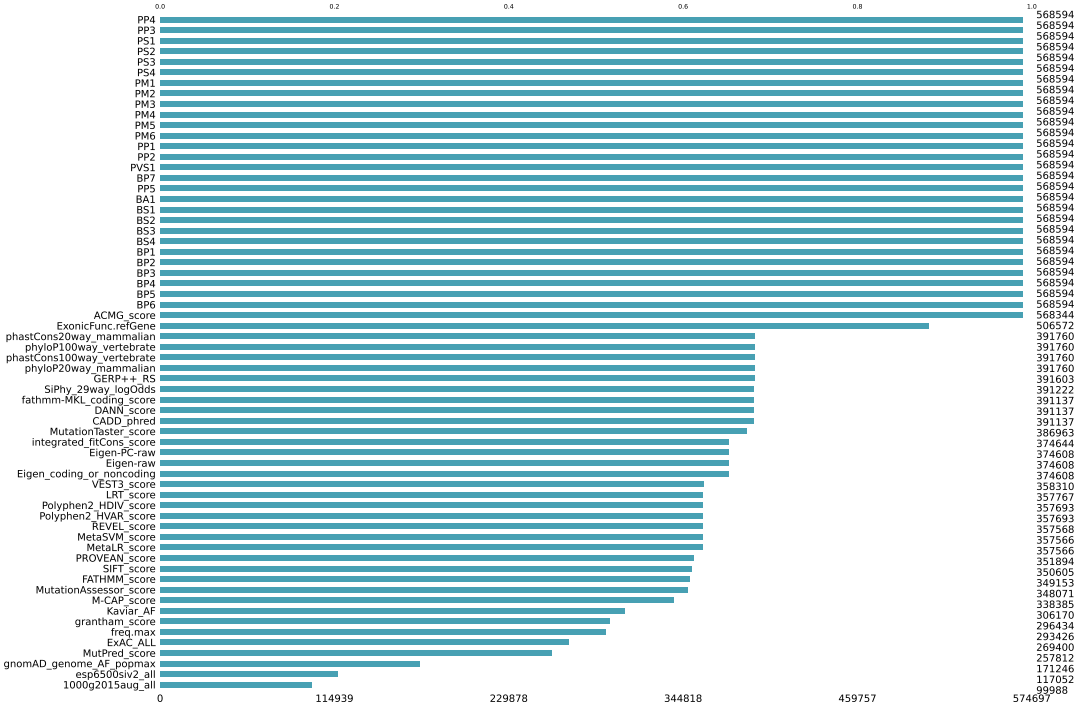


FIGURA 5.2: Diagrama de barras representando la completitud del conjunto de datos.

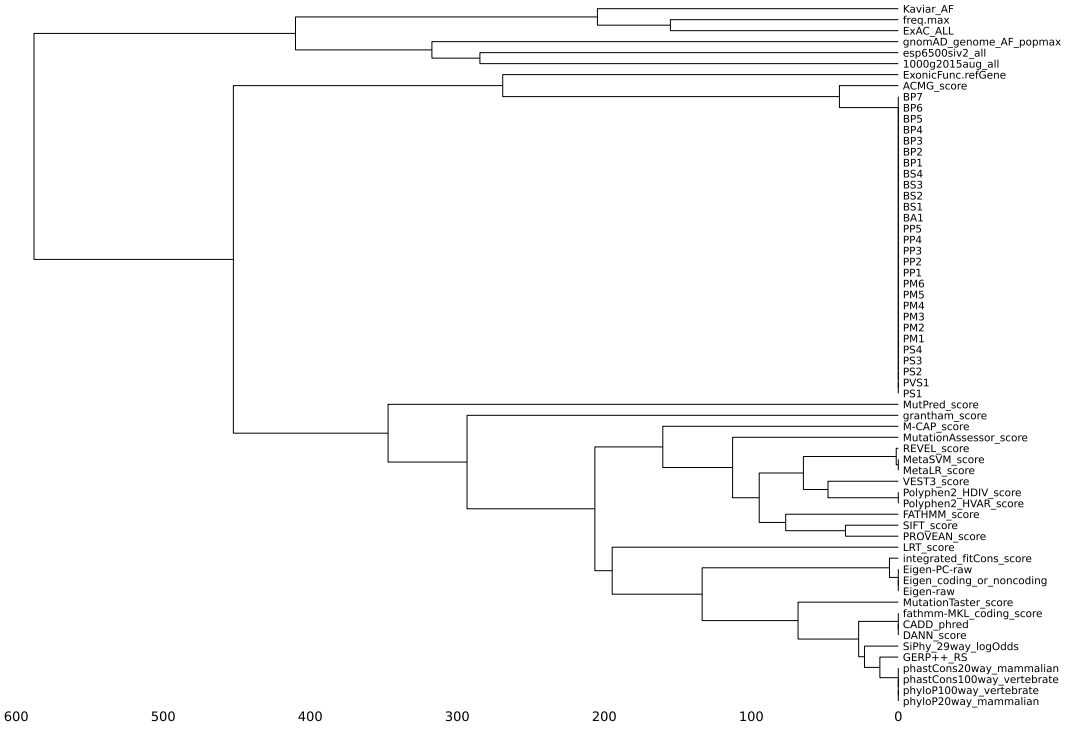


FIGURA 5.3: Dendrograma que representa la correlación entre las variables dados los patrones de valores faltantes.

Tipo	Cantidad antes	Cantidad después
Benigna	52527	21059
Probablemente Benigna	145877	19896
Patogénica	43492	39731
Probablemente Patogénica	22626	21424
VUS	310175	287337
Total	574697	**389447**

CUADRO 5.4: Distribución de tipos de variantes según su significado clínico o etiqueta de patogenicidad utilizadas en para el entrenamiento, luego de aplicados filtros respecto a valores faltantes.

5.3.2. Análisis de características

Para evaluar la correlación entre las características consideradas para el entrenamiento de los modelos, se puede observar la matriz de correlación visualizada en la Figura 5.4. Para esta representación se tuvo en cuenta previamente la eliminación de 7 características con varianza nula. Como puede observarse, en general aquellas características que muestran una correlación más elevada son las frecuencias poblacionales presentes y los predictores *in sílico*. En el caso de las frecuencias, por ejemplo, la frecuencia máxima se calcula a partir de las otras por lo que la observación cobra sentido. En el caso de los predictores, aquellos meta predictores que toman la información de otros predictores, como la conservación evolutiva, entre las variables usadas para generar su predicción son los que se encuentran altamente correlacionados con estos últimos. También se ven patrones entre algunas características y las evaluaciones de reglas ACMG/AMP: por ejemplo, hay una correlación considerable entre BA1 y la frecuencia alélicá máxima (BA1 depende de la frecuencia estrictamente) y PP3 con predictores *in sílico* (PP3 depende de la presencia de múltiples líneas de evidencia computacional que soportan el efecto deletéreo de una variante). Las características que presentan más de un 95 % de correlación son MetaSVM y MetaLR, y Eigen-raw y Eigen-PC, que pareados se tratan del mismo predictor pero con diferentes métodos. Estos son eliminados del conjunto total de características, además de algunos predictores de conservación evolutiva. A partir de este análisis se obtiene un total de 50 características finales para el conjunto de datos.

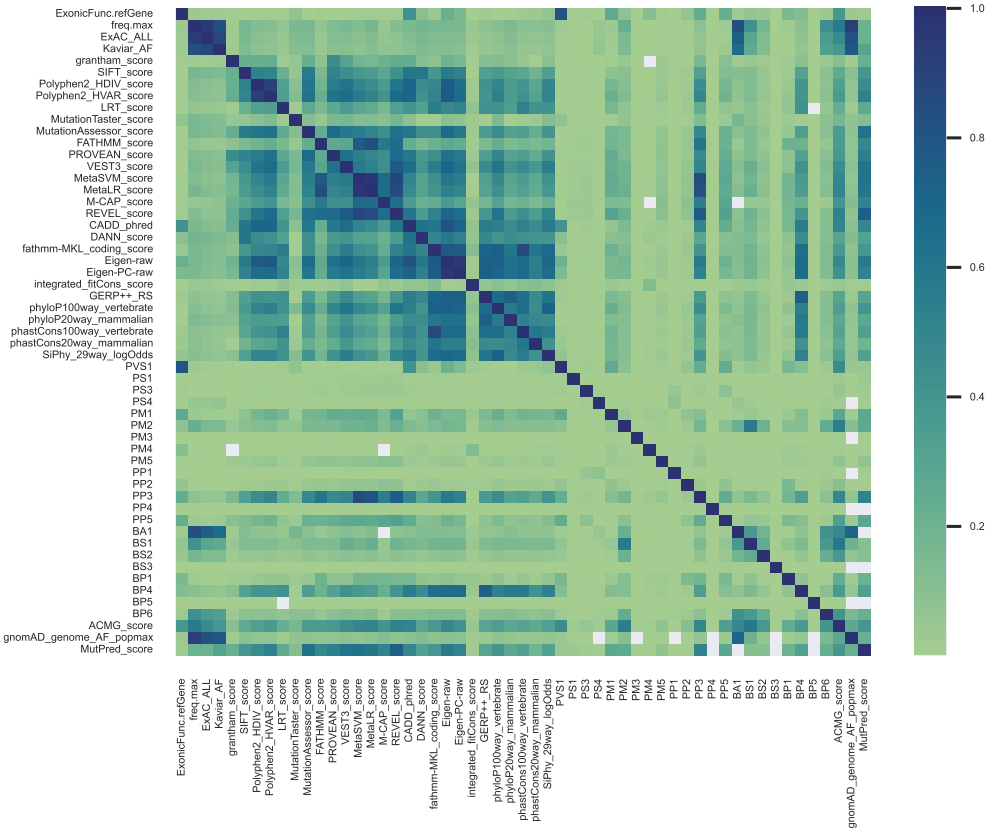


FIGURA 5.4: Correlación de variables finales consideradas para el entrenamiento de los modelos. Para este análisis fueron seleccionadas todas aquellas variables que presentaban varianza no nula.

5.3.3. Entrenamiento

Para el entrenamiento de los distintos modelos se presentarán los resultados obtenidos en la instancia de clasificación en 5 clases (B, LB, P, PP, VUS), teniendo en cuenta la separación de datos de entrenamiento y validación de forma aleatoria, a través de validación cruzada. Para esto se tomó un 80 % del conjunto de datos para entrenar el modelos, y un 20 % como prueba. El conjunto de datos de prueba no fue seleccionado de forma aleatoria, a los efectos de conseguir un conjunto de datos que no se encuentren en el conjunto de entrenamiento a través de contaminación mediante predictores. En la Tabla 5.5 se presenta el resumen de las cantidades de variantes de cada tipo correspondiente a cada conjunto de entrenamiento.

Significado Clínico	**Conjunto de entrenamiento**	**Conjunto de prueba**
Benigno	16847	4212
Probablemente benigno	15916	3980
Patogénico	16000	4000
Probablemente patogénico	17139	4285
VUS	16000	4000
Total	**53102**	**20477**

CUADRO 5.5: Cantidad de variantes seleccionadas para los conjuntos de entrenamiento y prueba para cada una de las 5 clases evaluadas.

Para estos conjuntos de datos se aplicaron los algoritmos mencionados anteriormente, los cuales fueron evaluados mediante el uso de validación cruzada. Los resultados de desempeño de los modelos se presentan en la Tabla 5.6.

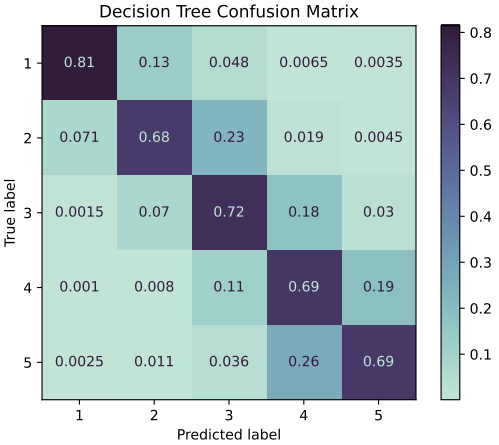
Modelo	***Train accuracy***	***Test accuracy***	**F1 macro**
KNN	$0{,}65 \pm 0{,}00$	0.66	$0{,}65 \pm 0{,}00$
Naive Bayes	$0{,}49 \pm 0{,}01$	0.20	$0{,}44 \pm 0{,}03$
SVM	$0{,}69 \pm 0{,}00$	0.69	$0{,}69 \pm 0{,}00$
Decision Tree	$0{,}70 \pm 0{,}00$	0.68	$0{,}70 \pm 0{,}00$
Random Forest	$0{,}78 \pm 0{,}01$	0.76	$0{,}78 \pm 0{,}00$
XGBoost	$0{,}78 \pm 0{,}00$	0.75	$0{,}79 \pm 0{,}0$

Cuadro 5.6: Desempeño de los clasificadores evaluados inicialmente.

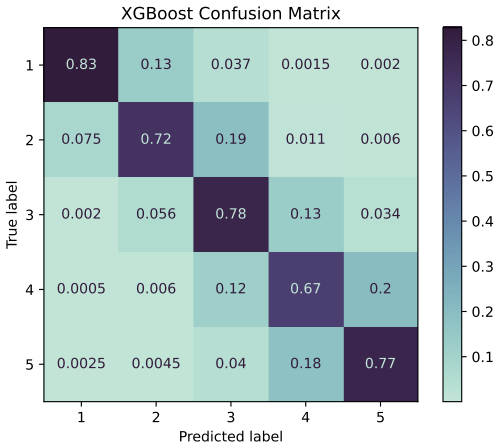
Como se puede observar en la tabla 5.6, el mejor desempeño se dio en *Decision Tree*, *Random Forest* y *XGboost*. Si bien la elección inicial de modelos se destinó a probar métodos clásicos de uso en aprendizaje automático, no fue arbitraria la selección de distintos modelos basados en modelos basados en árboles de decisión para las pruebas realizadas. Se ha demostrado en distintos trabajos que este tipo de modelos tiene un muy buen desempeño en problemas que se basan en variantes genómicas [4]. Estos modelos en general tienen una serie de características que resultan de interés para diversos problemas, y en específico en la clasificación de variantes. Por un lado, pueden proporcionar una representación clara del proceso de toma de decisiones que utilizó el modelo para realizar predicciones y la identificación de las características más importantes que contribuyeron a la predicción, lo que resulta un facilitador a la hora de la comprensión e interpretación de las predicciones de los modelos. Por otro lado, los modelos basados en árboles de decisión y ensamble pueden manejar de forma eficaz datos complejos y de alta dimensionalidad, permitiendo capturar las interacciones complejas entre las características. Por ejemplo, GBDT es muy robusto a la multicolinearidad cuando las característica son redundantes y se encuentran con una alta correlación. Una vez que se obtuvieron los mejores desempeños, pasaron evaluarse los 3 modelos mencionados anteriormente a los efectos de identificar los mejores hiperparámetros, e identificar el modelo final a usar.
El resultado obtenido en KNN muestra que el modelo no tiene un buen desempeño en el conjunto de datos en comparación con el resto de los métodos, a pesar de mostrar un desempeño uniforme a lo largo del conjunto de datos. *Naive Bayes* muestra los menores desempeños, además de presentar sobreajuste en los datos, lo que indica que en este caso particular, el modelo es el menos efectivo. Se puede ver que los modelos con un F1 más grade son *SVM*, *Random Forest*, *GBDT* a través de XGBoost, y *Decision trees*, lo que podría indicar que presentan un buen balance predicción-sensibilidad en los resultados de clasificación. En el caso de *Random Forest*, *GBDT*, y *Decision trees*, los resultados indicarían que funcionan de forma consistente en diferentes muestras de datos, y si bien tienen mucho para mejorar, lograrían predecir de forma correcta el resultado un porcentaje alto de los casos. No obstante, el *accuracy* en el conjunto de prueba indica que los 3 modelos no funcionan tan bien con datos no vistos previamente, lo que puede indicar sobreajuste. A continuación se seguirán evaluando estos 3 modelos con un mejor desempeño a través del ajuste de sus parámetros, para evaluar si es posible mejorar el rendimiento tanto en el conjunto de prueba, como en general.

A continuación se presentan los resultados de los 3 modelos diferentes ajustados mediante validación cruzada con *GridSearchCV* utilizando la métrica de precisión (*accuracy*) como función de evaluación. En esta prueba, para *Random Forest* se realizaron pruebas para los hiperparámetros `max_depth` y `n_estimators` utilizando los valores `[5, 10, 15]` y `[100, 200, 300]` respectivamente. Se encontró que la precisión en entrenamiento fue de $0{,}779 \pm 0{,}005$ mientras que la precisión en prueba fue de 0,758. Para *Decision Tree* se probaron los hiperparámetros `max_depth` y `min_samples_split`

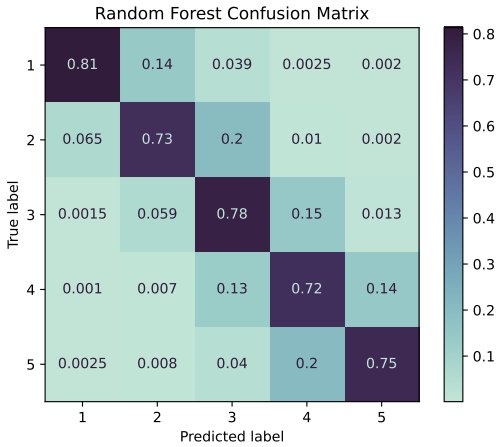
con los valores `[5, 10, 15]` y `[2, 5, 10]` respectivamente. Los resultados muestran que la precisión en entrenamiento fue de $0{,}740 \pm 0{,}007$ mientras que la precisión en prueba fue de 0.719. Por último, para `XGBClassifier` se probaron los hiperparámetros `max_depth` y `learning_rate` con los valores `[5, 10, 15]` y `[0.1, 0.01, 0.001]` respectivamente. La precisión en entrenamiento y en prueba fueron $0{,}788 \pm 0{,}005$ y 0,754 respectivamente. En el caso de `XGBClassifier`, los resultados de media y desviación indican que el modelo tiene un funcionamiento razonable con los datos de entrenamiento. La precisión en los datos prueba de 0.754 si bien no es mala, indica que el modelo no funciona tan bien en los datos de prueba como en los de entrenamiento. Esto podría deberse a un exceso de ajuste debido a la complejidad del modelo. La matriz de confusión (Figura 5.5b) nos permite ver que el modelo está haciendo las predicciones más correctas para la clase B (1) y PP (4). Sin embargo, tiene más dificultades con PB (2) y P (5), ya que hay un número relativamente alto de falsos positivos y falsos negativos en estas clases. La clase VUS (3) también parece problemática, ya que hay un número relativamente alto de falsos positivos y falsos negativos en esta clase también. Dada la naturaleza de la clasificación, tiene sentido que sea difícil diferenciar una VUS de los otros pasos, especialmente de PB y PP. Respecto a *Decision Tree* la matriz de confusión muestra que el modelo tiene más dificultades para clasificar correctamente las variantes PP, donde la precisión es la más baja (0.60), y en donde se muestra que hay un mayor desempeño en general. En general el mayor desempeño también es en la clasificación de variantes B, teniendo una mayor precisión y sensibilidad en este caso. El modelo tiene un rendimiento relativamente aceptable de precisión de 0.719, sin embargo, aún puede mejorar, especialmente en la clasificación correcta de las muestras de determinadas clases. Por último, y en comparación con los dos anteriores, todo indica que *Random Forest* (Figura **??** tiene la mayor precisión en datos de prueba, con 0,758, sin embargo, cabe señalar que los modelos de*Random Forest* y `XGBClassifier` tienen intervalos de confianza superpuestos con sus precisiones de entrenamiento, mientras que el modelo de *Decision Tree* tiene una precisión de entrenamiento más baja y un intervalo de confianza más amplio. Si observamos las matrices de confusión para *Random Forest* (Figura 5.5c) y podemos ver que los tres modelos tienen un rendimiento similar en las distintas clases, con algunas variaciones en la precisión y sensibilidad. En general, el rendimiento de los modelos parece adecuado, pero puede haber margen de mejora para clasificar ciertas clases con mayor especificidad. Es necesario hacer énfasis en los falsos positivos en casos opuestos, y en la diferencia de su impacto. Teniendo en cuenta que el objetivo final (no de este trabajo) es asistir un diagnóstico, los falsos negativos en la clase patogénica pueden ser graves, no tanto si se clasifican como PP, sino como B o PB. En general la mayor diferencia se encuentra entre el conjunto Variantes B-PB y P-PP. Esto es esperado y un buen indicio, ya que dentro de lo que los modelos puedan no acertar, es importante diferenciar aquellas variantes que son o podrían ser patogénicas de las que son o podrían ser benignas. Como se puede ver en la matriz de confusión para el modelo que tuvo una mejor precisión, hay una baja tasa en donde se "confunden" variantes de tipo Patogénico con Benigno, lo que es una buena base para seguir explorando, más allá de las debilidades de los modelos.



(A) Matriz de confusión para el mejor modelo logrado con *Decision Trees*.



(B) Matriz de confusión para el mejor modelo logrado con *XGBoost*



(C) Matriz de confusión para el mejor modelo logrado con *Random Forest*

FIGURA 5.5: Matrices de confusión de los modelos finales obtenidos para la clasificación de 5 clases: B(1), LB(2), VUS(3), LP(4), P(5)

En general, según los resultados proporcionados, parece que el modelo *Random Forest* es el que obtiene mejores resultados en términos de precisión, tanto en los conjuntos de entrenamiento como en los de prueba. Tiene la mayor precisión de prueba y también la menor diferencia entre la precisión de entrenamiento y la de prueba (como indica la menor desviación estándar en la precisión de entrenamiento), lo que sugiere que no se está ajustando en exceso a los datos de entrenamiento. Los otros dos modelos, GBDT y *Decision trees*, tienen menores precisiones en las pruebas y mayores desviaciones estándar en sus precisiones de entrenamiento. Esto indica que pueden estar sobre-ajustándose a los datos de entrenamiento, lo que conduce a un menor rendimiento de generalización en datos no vistos. Que los modelos logren generar buenas predicciones sobre el conjunto de entrenamiento pero no en los conjuntos nuevos o no vistos previamente, en el problema actual, no es un aspecto menor, dado el efecto que pueda tener sobre la clasificación real de una variante, asociada a un diagnóstico. Si bien GBDT está siendo ampliamente utilizado en la literatura, específicamente en el campo de estudio dada la potencia de *XGBoost*, los aspectos mencionados anteriormente, en el trabajo actual, ponen a *Random Forest* por encima de GBDT, siendo, en esta instancia, el modelo seleccionado para las

pruebas iniciales.
En general, los datos genómicos son complejos, lo que traduce en la dificultad la construcción de modelos precisos. Además, puede haber variables de confusión que no se tengan en cuenta en los datos, lo que también puede afectar al rendimiento de los modelos. Por lo tanto, es importante preprocesar y analizar cuidadosamente los datos. El pre-procesamiento puede definir completamente el curso de los modelos, y se considera que es un aspecto a trabajar y mejorar ampliamente en el presente trabajo. Como fue mencionado anteriormente, el presente trabajo implicó el desarrollo de distintos tipos de pruebas, tanto en el pre-procesamiento como en las combinaciones de modelos e hiperparámetros seleccionados. En este caso, se ha presentado una pequeña parte de esta exploración, que es la que arroja los resultados más desafiantes. En evaluaciones previas, se ha trabajado con la clasificación en distintos grupos de clases, los que en general arrojaron, con los mismos modelos seleccionados como aquellos con un mejor desempeño, resultados distintos y en general mejores en cuanto a precisión, sensibilidad y especificidad. Esto se trata del trabajo en clases binarias (B-P), y 4 clases (B, PB, P, PP). En el caso de la clasificación binaria, *Random Forest* se aplicó en un conjunto de variantes donde con el uso de 63 características se obtuvo una precisión en entrenamiento y prueba de 0.99. No obstante, se identificaron problemas de contaminación entre los conjuntos de prueba y entrenamiento. En pruebas posteriores con 4 clases, con las mismas características y modificando el pre-procesamiento y separación de datos se obtuvo una precisión de 0.98 en datos de entrenamiento, y 0.81 en datos de prueba, lo que impulsó a ahondar más en esta separación de datos y en la clasificación multi-clase. El último resultado de estas pruebas con 4 clase simplicó el uso de 41 características, con un desempeño de *Random Forest* en 0.9 en datos de entrenamiento y 0.86 en datos de prueba. Todas estas pruebas realizadas anteriormente se hicieron en distintas instancias del conjunto de dato, y lo que se refleja en los datos mostrados anteriormente es el procesamiento más actual realizado sobre los datos, que implicó el refinamiento más detallado de las características y el conjunto de datos en general, no obstante, como se mencionó anteriormente, se considera que esta instancia es la que mayor complejidad ofrece a este tipo de datos, por lo que resta mucho trabajo por hacer. El objetivo de lograr separar las 5 clases, tiene una motivación clara en base a las reglas ACMG, no obstante, como se observó, es la separación que más cuesta lograr, y sobre la cual aún resta mucho trabajo.

5.4. Conclusiones y trabajo futuro

En esta instancia actual del trabajo no se buscó generar un clasificador que mejore la *performance* que tienen otros predictores similares, sino lograr una exploración de los mismos, como una forma de identificar la complejidad del problema, además de tener un acercamiento inicial en algoritmos de clasificación. Los resultados obtenidos no llegan a alcanzar el desempeño de los predictores existentes, lo que puede estar dado por varios factores (condiciones usadas, datos usados, características, etc), pero indica que aún resta trabajo por hacer. Se puede decir igualmente que se parte de una buena base para aplicarlo, o para mejorar lo implementado en una futura aplicación, no solo desde el punto de vista de los modelos ya implementados, sino también de las herramientas adquiridas. El objetivo de obtener un clasificador que logre discriminar los datos se ha cumplido parcialmente dada que la precisión aún no se considera aceptable para la aplicación esperada.

Más allá de que era esperable en base a la literatura que los métodos de agregación funcionen con un mejor desempeño en los datos (que es a lo que se llegó), es necesario ampliar el estudio realizado en cada método, y realizar la búsqueda de los parámetros óptimos en cada uno de ellos. Si bien en este trabajo no se realizaron más que pruebas iniciales con redes neuronales, se puede ampliar el trabajo en este campo también a los efectos de buscar mejorar el desempeño.

Otro aspecto interesante a analizar a futuro es la importancia de las variables, y la exploración de distintas combinaciones de predictores y características. La generación de características propias, o la identificación de otras fuentes de anotación, es uno de los aspectos cruciales para trabajar a futuro, que puede implicar la construcción de las reglas ACMG/AMP, o el uso de herramientas más actuales

que las calculen (sin restricciones de acceso), además de la incorporación de aspectos fenotípicos. También es importante considerar aspectos que reflejen el comportamiento normal de la clasificación de variantes, como la re-evaluación de las mismas, para lo que puede haber un potencial interesante en la revisión de literatura mediante herramientas de NLP.
Por otro lado, el trabajo en la selección correcta de conjuntos de entrenamiento y prueba es un aspecto no menor, y que pone a esta aplicación en dificultades constantes: los predictores usados para entrenar el modelo usando los datos de variantes existentes para brindar sus valores. Esto hace que haya cierta contaminación, y ya se ha observado su efecto en la reducción de la *performance* en la priorización de genes nuevos asociados a enfermedades raras.
Otro aspecto importante a considerar a futuro es la posibilidad de aplicar criterios más restrictivos a los datos utilizados, por ejemplo, al momento de elegirlos, considerar el soporte o curación que tienen las variantes, es decir, si hay sustento de su significado clínico, y el nivel de sustento que existe.
Como ya fue mencionado en ocasiones anteriores, aún resta establecer el vínculo entre las herramientas de clasificación y la plataforma, desde el punto de vista de los datos. El interés principal es lograr un clasificador que contemple la participación de expertos desde distintos aspectos. Inicialmente, se espera que la integración se haga desde la generación de etiquetas consenso. Como fue mencionado anteriormente, queda planteado como trabajo a futuro la implementación de estrategias que al momento de generar un consenso de múltiples expertos puedan contemplar el proceso heterogéneo de asignación de etiquetas. En este sentido, trabajos que estén orientados a generar modelos de clasificación para la generación de etiquetas consenso, y que logren modelar la decisión de los expertos de forma individual, puede resultar un buen comienzo [271]. A partir de allí, se podrían identificar qué características pueden aportar a mejorar el modelo, y representar de forma confiable la participación humana en el proceso.
Respecto a los objetivos planteados en esta instancia, se puede decir que se cumplieron, al igual que en el capítulo anterior, en una fase exploratoria, que más allá de no haber logrado un método de clasificación con un desempeño bueno, permite un primer acercamiento a la discriminación de variantes, sobre el cual hay que seguir trabajando.

Capítulo 6

Conclusiones finales

El presente trabajo implicó el desarrollo de las instancias iniciales de un sistema de clasificación de variantes. Si bien ya existen sistemas con estas características, se considera que la potencialidad de un sistema completo y bien implementado, de código abierto, gratuito y a nivel local y regional, puede tener un impacto positivo. Como se observó en casa instancia de desarrollo, de forma individual, pero también en conjunto cada parte está "en pañales", no solo desde un punto de vista tecnológico y de las mejoras que hay que hacer para que funcione como un proyecto global inicial, sino en todas las mejoras que se pueden hacer. No obstante, a través del desarrollo de este proyecto, se pudo llegar a un primer acercamiento a lo que implica la implementación de herramientas de clasificación de variantes, en varios aspectos: desde un punto de vista automático, como también manual. Y en el caso manual, no solo se logró conocer el proceso, sino que se pudo tener un primer acercamiento a cómo este proceso puede entrenar a usuarios. No hay parte del proyecto generado que no tenga aspectos a mejorar de forma amplia, por lo que su desarrollo es un puntapié interesante a muchos proyectos pequeños que aporten al desarrollo global de la plataforma. En función de los objetivos, se logró la exploración de herramientas de aprendizaje automático, y la obtención de un mecanismo para discriminar variantes en las 5 categorías establecidas por ACMG/AMP, a través de la obtención de conjuntos de datos de diversas fuentes y bases de datos públicas, y su integración preliminar a una plataforma de prestaciones básicas que permite que los usuarios clasifiquen variantes con un significado no definido, y se entrenen en la tarea. La integración de los dos desarrollos y los resultados asociados a las mismas son una gran parte del trabajo por hacer, para lo que se requerirá del trabajo de varias partes para lograrlo.

Apéndice A

Reglas ACMG

A.1. Criterios de clasificación de las variantes patogénicas

Evidencia muy fuerte de patogenicidad	
PVS1	Null variant (*nonsense*, *frameshift*, canónica +/-1 o 2 sitios de empalme, codón de iniciación, deleción de uno o varios exones) en un gen con pérdida de función (LOF) es un mecanismo conocido de enfermedad.
Evidencia fuerte de patogenicidad	
PS1	Mismo cambio de aminoácido que una variante patogénica previamente establecida independientemente del cambio de nucleótidos.
PS2	De novo (tanto maternidad como paternidad confirmadas) en un paciente con la enfermedad y sin antecedentes familiares.
PS3	Estudios funcionales in vitro o in vivo bien establecidos que apoyen un efecto perjudicial en el gen o producto génico.
PS4	La prevalencia de la variante en los individuos afectados es significativamente en comparación con la prevalencia en los controles.
Evidencia moderada de patogenicidad	
PM1	Situado en un *hot stpot* mutacional y/o dominio funcional crítico y bien establecido (por ejemplo, el sitio activo de una enzima) sin variación benigna.
PM2	Ausente en los controles (o con una frecuencia extremadamente baja si es recesivo) en Exome Sequencing Project, 1000 Genomes o ExAC
PM3	Para trastornos recesivos, detectados en trans con una variante patogénica.
PM4	Cambios en la longitud de la proteína debidos a deleciones/inserciones dentro del marco en una región no repetida o variantes de pérdida de codón *stop*.
PM5	Nuevo cambio *missense* en un residuo de aminoácido en el que se ha observado antes un cambio *missense* diferente determinado como patógénico.
PM6	Asunción de novo, pero sin confirmación de paternidad y maternidad.
Evidencia de patogenicidad de soporte	
PP1	Co-segregación con enfermedad en múltiples familiares afectados en un gen que se sabe definitivamente que causa la enfermedad.
PP2	Variante *missense* en un gen que tiene una tasa baja de variación *missense* benigna y donde las variantes *missense* son un mecanismo común de enfermedad.
PP3	Múltiples líneas de evidencia computacional apoyan un efecto nocivo sobre el gen o el producto génico (conservación, evolución, impacto del empalme, etc.).
PP4	El fenotipo del paciente o los antecedentes familiares son altamente específicos para una enfermedad con una etiología genética única.
PP5	Una fuente reputada informa recientemente de que la variante es patogénica, pero el laboratorio no dispone de pruebas para realizar una evaluación independiente.

Cuadro A.1: Lista de criterios para la clasificación de variantes patogénicas. Extraído, traducido y modificado de [44].

A.2. Criterios de clasificación de las variantes benignas

Evidencia *stand-alone* de impacto benigno	
BA1	La frecuencia alélica es superior al 5 % en Exome Sequencing Project, 1000 Genomes, o ExAC.
Evidencia fuerte de impacto benigno	
BS1	La frecuencia alélica es mayor de lo esperado para el trastorno
BS2	Observado en un individuo adulto sano para un trastorno recesivo (homocigoto), dominante (heterocigoto) o ligado al cromosoma X (hemicigoto) con completa a una edad temprana.
BS3	Los estudios funcionales in vitro o in vivo bien establecidos no muestran ningún efecto perjudicial en la función o el empalme de proteínas.
BS4	Falta de segregación en los miembros afectados de una familia
Evidencia de impacto benigno de soporte	
BP1	Variante *missense*en un gen para el que se conocen que variantes principalmente truncantes causan enfermedades.
BP2	Observado en trans con una variante patogénica para un gen/trastorno dominante totalmente penetrante; u observado en cis con una variante patogénica en cualquier patrón hereditario.
BP3	Deleciones/inserciones en una región repetitiva sin función conocida.
BP4	Múltiples líneas de evidencia computacional sugieren que no hay impacto en el gen o producto génico (conservación, evolución, impacto del empalme, etc.)
BP5	Variante encontrada en un caso con una base molecular alternativa para la enfermedad.
BP6	Fuente reputada informa recientemente de la variante como benigna pero las pruebas no están a disposición del laboratorio para realizar una evaluación independiente.
BP7	Una variante sinónima (silenciosa) para la que los algoritmos de predicción de *splicing* predicen que no afectará a la secuencia consenso de empalme ni creará un nuevo sitio de empalme Y el nucleótido no está altamente conservado

Cuadro A.2: Lista de criterios para la clasificación de variantes benignas. Extraído, traducido y modificado de [44].

A.3. Reglas de combinación de criterios

Patogénica	
1	1 Very Strong (PVS1) AND a. ≥ 1 Strong (PS1–PS4) OR b. ≥ 2 Moderate (PM1–PM6) OR c.1 Moderate (PM1–PM6) and 1 Supporting (PP1–PP5) OR d. ≥ 2 Supporting (PP1–PP5)
2	≥ 2 Strong (PS1–PS4) OR
3	1 Strong (PS1–PS4) AND a. ≥ 3 Moderate (PM1–PM6) OR b. 2 Moderate (PM1–PM6) AND ≥2 Supporting (PP1–PP5) OR c.1 Moderate (PM1–PM6) and 1 Supporting (PP1–PP5) OR d. 1 Moderate (PM1–PM6) AND ≥4 Supporting (PP1–PP5)
Probablemente Patogénica	
1	1 Very Strong (PVS1) AND 1 Moderate (PM1–PM6) OR
2	1 Strong (PS1–PS4) AND 1–2 Moderate (PM1–PM6) OR
3	1 Strong (PS1–PS4) AND ≥2 Supporting (PP1–PP5) OR
4	≥3 Moderate (PM1–PM6) OR
5	2 Moderate (PM1–PM6) AND ≥2 Supporting (PP1–PP5) OR
6	1 Moderate (PM1–PM6) AND ≥2 4 Supporting (PP1–PP5)
Benigna	
1	1 Stand-Alone (BA1) OR
2	≥2 Strong (BS1–BS4)
Probablemente Benigna	
1	1 Strong (BS1–BS4) and 1 Supporting (BP1–BP7) OR
2	≥2 Supporting (BP1–BP7)

Cuadro A.3: Reglas de combinación de criterios para clasificar variantes. Extraído de [44].

Apéndice B

Datos

B.1. Formato VCF

El formato VCF es un formato de texto utilizado para almacenar información de variaciones en la secuencia de ADN con respecto a una secuencia de referencia. En la Figura B.1 se presenta un ejemplo de la estructura de un archivo VCF en su versión 4.0.

```
##fileformat=VCFv4.2
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens",taxonomy=x>
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF    ALT     QUAL FILTER INFO                              FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257 G      A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:.
20     17330   .         T      A       3    q10    NS=3;DP=11;AF=0.017               GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3   0/0:41:3
20     1110696 rs6040355 A      G,T     67   PASS   NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2   2/2:35:4
20     1230237 .         T      .       47   PASS   NS=3;DP=13;AA=T                   GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
20     1234567 microsat1 GTC    G,GTCT  50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

FIGURA B.1: Ejemplo de estructura de un archivo VCF. Extraído de [272].

El líneas generales, un archivo VCF contiene líneas de meta-información, un encabezado y líneas de datos que contienen información sobre posiciones en las que se encuentran variantes en el genoma. Las líneas de meta-información describen las entradas de INFO, FILTER y FORMAT que se encuentran en el cuerpo del VCF (Figura B.1). La línea que corresponde al encabezado consiste en una línea delimitada por tabulaciones que nombra a las 8 columnas fijas de todo archivo VCF: CHROM, POS, ID, REF, ALT, QUAL, FILTER, INFO. Finalmente, las líneas de datos, también delimitadas por tabulaciones, contienen información asociada a las variantes, separada en los 8 campos mandatorios [272]:

- CHROM (cromosoma): representa un identificador del genoma de referencia, que apunta al *contig* en el archivo de ensamblaje. Debe contener un *string* alfanumérico.
- POS (posición): contiene la posición de referencia, en la que la primer base en la variante corresponde a la posición 1. Este campo debe contener un número entero.
- ID (identificador): corresponde a una lista de identificadores. Si la variante se encuentra en dbSNP, en este campo se pueden utilizar el o los números de rs (rsID), es decir el número de acceso utilizado por los investigadores y las bases de datos para referirse a SNPs específicos. Debe contener un *string* alfanumérico.

- REF (base(s) de referencia): contiene la o las bases que están en el genoma de referencia y se ven cambiadas en el genoma en cuestión. Cada base puede ser A, C, G, T o N. Este campo debe contener un contener un *string*.
- ALT (base alternativa): contiene una lista de los alelos que no son de referencia. Cada base puede ser A, C, G, T o N. Este campo debe contener, al igual que REF, un *string*.
- QUAL (calidad): representa el *score* de calidad de *Phred* para la afirmación realizada en ALT. Es decir, que si Q representa el *Prhed score*, $Q = -10 \log_{10} P$, donde P es la probabilidad de haber llamado incorrectamente a la base ALT. Este campo debe contener un valor numérico.
- FILTER (estado del filtrado): contiene el *string* PASS si la posición ha logrado pasar todos los filtros.
- INFO (información adicional: contiene campos codificados como una serie de claves separadas por punto y coma con valores opcionales en el formato: <clave>=<datos>[,datos]. Si bien se permite el agregado de claves arbitrarias, hay ciertos sub-campos que se encuentran reservados y son a su vez opcionales. Algunos de los sub-campos presentes en los datos obtenidos de 1000 genomas son los siguientes:
  - AA: alelo ancestral.
  - AC: recuento de alelos en los genotipos para cada alelo ALT.
  - AF: frecuencia alélica estimada para cada alelo ALT en el rango (0,1).
  - AN: número total de alelos en los genotipos.
  - NS: número de muestras con datos.
  - DP: profundidad de lectura total.
  - VT: tipo de variante representada en la línea.

  entre otros sub-campos.

Apéndice C

Plataforma web

C.1. Base de Datos

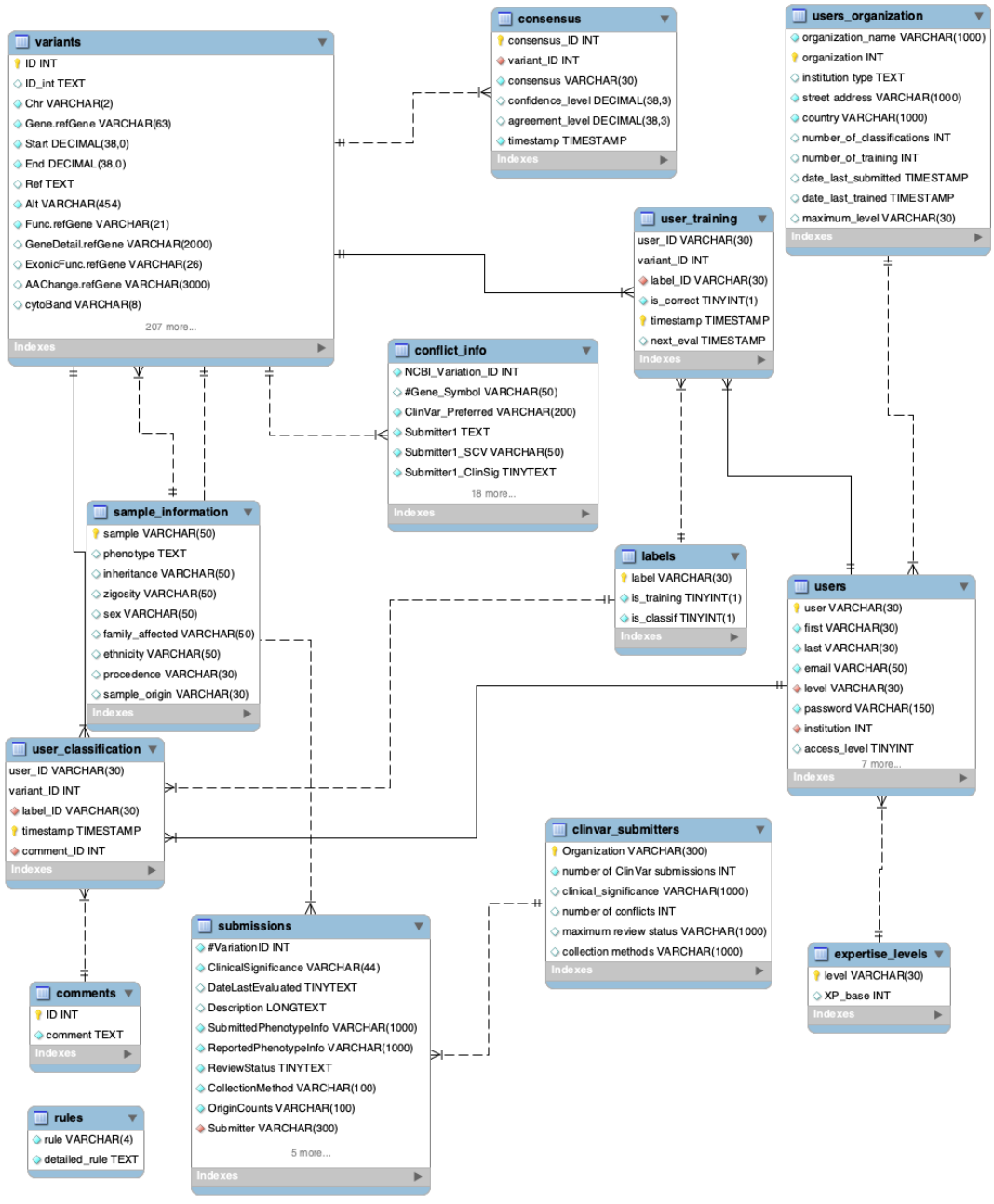


FIGURA C.1: Diagrama EER (*Enhaced entity-relationship*, en inglés) de la base de datos de la plataforma.

C.2. Evaluación

C.2.1. Metodología de evaluación

C.2.1.1. Experimento 1: Definición de información a mostrar

- Hipótesis: el uso de la plataforma aporta un diferencial cualitativo en la información presentada.
- Público objetivo: Experto asesor, avanzado asesor.
- Alcance: Evaluación de los datos incluidos en la anotación, priorización de las columnas que son las más frecuentes en su uso para su mostrado en la interfaz, evaluación de otras características a agregar.
- Definición de éxito (*feedback* cualitativo):
  - La información mostrada es efectiva: se toma como métrica de éxito la reducción de la información a lo estrictamente necesario, observando que se puede realizar la tarea con la misma calidad.
  - Se agregó información relevante: Se agregan datos que aportan nuevos puntos de vista a la clasificación.

C.2.1.2. Experimento 2: Organización de contenido y presentación de la variante

- Hipótesis: La forma en la que se dispone la información es propicia para incorporar el proceso de clasificación de variantes.
- Público objetivo: Experto asesor, avanzado asesor.
- Alcance: Identificación de la forma de exponer la información al usuario para que la evaluación se haga ordenada, obtener una categorización completa de los datos seleccionados en el experimento anterior, estudiar un posible agrupamiento en pestañas, evaluar formas de visualización distintas al formato tabular para aquellas características que sean adecuadas
- Definición de éxito(*feedback* cualitativo):
  - La información se muestra de forma ordenada y bien agrupada: la categorización y orden de los datos refleja el flujo de trabajo a realizar.
  - La información se muestra de forma "amigable": Más allá de los aspectos que se puedan mejorar en relación al diseño de la página, la misma muestra la información de forma concisa e intuitiva, y se presenta una mejora cualitativa respecto al relevamiento de la información en una planilla (método que se usa regularmente para evaluar la patogenicidad de variantes).

C.2.1.3. Experimento 3: Ingreso de nuevas variantes a la plataforma

- Hipótesis: el usuario puede incluir nuevas variantes para su estudio y/o almacenamiento.
- Público objetivo: Experto asesor, avanzado asesor.
- Alcance: Evaluación de las columnas sugeridas al usuario, evaluación de la posibilidad de incluir variantes a través de un VCF, el cual debe ser anotado posteriormente.
- Definición de éxito (*feedback* cualitativo):
  - El usuario puede ingresar datos nuevos: La variante ingresada puede ser clasificada.

- La información mínima solicitada es abarcable por el usuario: No debería presentar dificultades extra ni procesamiento el ingreso de una variante más que lo que el usuario accede de una variante normalmente.
- La información de la variante puede ser ampliada: Los métodos de ampliación de la información permiten obtener variables "clasificables´´ desde un punto de vista de completitud de la información.

C.2.1.4. Experimento 4: Evaluación de paciente para comparar con planilla

- Hipótesis: Usar la plataforma mejora la experiencia de clasificación de variantes
- Público objetivo: Experto asesor, avanzado asesor.
- Alcance: Evaluación de la experiencia conjunta del resultado de la información obtenida y su organización (puntos 1 y 2 evaluados anteriormente) en su aplicación a datos de pacientes.
- Definición de éxito (*feedback* cualitativo):
  - El usuario logra evaluar las variantes de un paciente sin pérdida efectiva de información: se logra evaluar al menos la misma cantidad de variantes respecto a su evaluación en planilla, y la precisión se mantiene.
  - El usuario logra evaluar las variantes de un paciente con una mejora en su experiencia general: La organización, disposición y selección de información hacen que la experiencia de evaluación de la información sea más cómoda a su evaluación en una planilla.
  - En consecuencia de lo anterior, la evaluación podría hacerse más rápido que antes: la evaluación se logra hacer en un tiempo menor que el usual, y podría llegar a evaluarse un número mayor de variantes en un tiempo determinado.

C.2.1.5. Experimento 5: Evaluación de plataforma de etiquetado

- Hipótesis: La plataforma implementada es adecuada para la tarea de etiquetado manual de variantes.
- Público objetivo: Experto asesor, avanzado asesor.
- Alcance: Evaluación de la plataforma como un espacio acorde de etiquetado para su posterior procesamiento en cuanto a calidad de datos y etiqueta.
- Definición de éxito (*feedback* cualitativo):
  - Los insumos finales brindados como etiquetas son acordes para la tarea: desde el punto de vista de aporte a una clasificación automática el abordaje es correcto.
  - Las variantes que se ofrecen para brindar etiquetas son útiles: el conjunto de datos es representativo.

C.2.1.6. Experimento 6: Evaluación de plataforma inicial de aprendizaje

- Hipótesis: Usar la plataforma ayuda a entender el proceso de clasificación de variantes.
- Público objetivo: Todos los niveles.
- Alcance: Evaluar la experiencia en distintos niveles de usuario en cuanto al aprendizaje del proceso de clasificación.
- Definición de éxito (*feedback* cualitativo):

- Para los no-expertos, la organización de la información y su categorización es una ayuda al momento de aplicar reglas de clasificación.
- Es una gran ayuda para el/la principiante.
- No faltan aspectos a evaluar para el experto.

C.3. Evaluación de experiencia de usuarios preliminar

A los efectos de contar con una evaluación preliminar de parte de los usuarios, se realizó un relevamiento de sus opiniones respecto a aspectos de interés de la plataforma. En la Sección C.2 se presenta la metodología seguida para dicha evaluación. En esta instancia se obtuvo la opinión de 4 usuarios de la plataforma: 1 experto, 2 avanzados, 1 competente y 1 principiante, por lo que se considera una instancia preliminar de evaluación. Entre los usuarios evaluadores habían 2 asesores del desarrollo del trabajo: un experto y un avanzado. El usuario Competente, si bien no participó en el asesoramiento, pertenece al grupo de trabajo en el que se enmarca la plataforma. Además, se cuenta con la evaluación de un usuario avanzado y un principiante que son externos al grupo.
En la Tabla C.1 se presenta un resumen de las respuestas más importantes adquiridas para cada experimento.

Los aspectos valorados se refieren a la usabilidad de la plataforma en cuanto a: información disponible de las variantes, organización del contenido y funcionamiento de los casos de uso planteados. Las evaluaciones consistieron en general en presentar un conjunto de variantes seleccionado *ad hoc* para completar los distintos casos de uso, y luego obtener su opinión respecto al cumplimiento de las hipótesis planteadas. La valoración de los usuarios se adquiere a través de formularios, en donde las repuestas se brindan en una combinación de escala de tipo Lickert [273] de cinco niveles[1] (cuando se usan enteros), y respuestas desarrolladas. Las opiniones fueron brindadas luego de evaluar aproximadamente 10 variantes por cada experimento.

El primer experimento se refiere a la información presentada al usuario para que pueda evaluar una variante. Lo que se obtiene de parte de los usuarios participantes es su punto de vista respecto a la información que se muestra sin considerar aún su organización: si la información presentada es completa para la tarea que se quiere realizar, si la misma es adecuada en el sentido de clasificar variantes en un contexto del diagnóstico de enfermedades raras, si hay sugerencias o cambios que se consideren de relevancia a futuro. En la evaluación participaron usuarios de cada nivel (independientemente de que se esperaba evaluar principalmente a expertos y avanzados). Como se puede observar en la Tabla C.1 las respuestas tienden a valorar positivamente la información presentada. En el caso en que no se emitió una valoración, se trató de un usuario que no se consideraba con herramientas para diferenciar específicamente si se requiere más información. Las sugerencias recibidas respecto a información a agregar son acordes a los planteos realizados respecto a abordajes futuros. Más allá de que las variantes con las que se cuenta actualmente no ofrecen todas las características necesarias para lograr un veredicto sobre su patogenicidad, hay información que podrían adquirirse a través de la integración de fuentes adicionales a las trabajadas en el proyecto actual (como las sugeridas por el usuario avanzado). Por otro lado, hay información sugerida a agregar, como de estudios de asociación, a la que es posible acceder en la base de datos y agregar en la interfaz, pero que quedó por fuera en la priorización de características a mostrar. En líneas generales, al momento de emitir una variante, lo más destacado como información que se pierde respecto a un estudio regular refiere a calidad de la llamada, calidad de alelos, profundidad y cigosidad, a los efectos de permitir detectar posibles artefactos o unir información de la muestra respecto al modo de herencia de la variante en los fenotipos asociados.

[1] 1: Totalmente en desacuerdo, 2: En desacuerdo, 3: Ni de acuerdo ni en desacuerdo, 4: De acuerdo, 5: Totalmente de acuerdo.

El segundo experimento busca relevar la opinión respecto a cómo se dispone la información evaluada anteriormente. El planteo de la plataforma apuesta a que la misma debe permitir, a través de una visualización mínimamente acorde, la clasificación siguiendo reglas, y para confirmar la hipótesis, se evalúa si: el orden y la agrupación están bien logrados, y si la forma en la que se presenta la información ayuda a la experiencia del usuario comparado con otros métodos estándares como una planilla. En este sentido los resultados arrojan resultados también favorables, donde los usuarios valoran muy positivamente la disposición de la información. En este sentido, el punto más importante a considerar en mejoras futuras es la posibilidad me mejoras en la representación visual de la información, lo cual está abordad con recursos mínimos en esta instancia. Además, se puede brindar mayor flexibilidad en la búsqueda y acceso a las variantes, a los efectos de que si bien la interfaz mejora la experiencia, no se pierda la flexibilidad que un usuario tiene al observar los datos más crudos. Esto haría que la interfaz sea cómoda para quienes no están cerca de la manipulación de datos, pero para los que si también.

En el tercer experimento se evaluó la posibilidad de cargar nuevas variantes a la plataforma. Si bien se anticipó que el mecanismo funcional actual no es el más adecuado y directo para quien lo utiliza, se consideró importante obtener una percepción externa al respecto. En ese marco, la evaluación confirma, que más allá de la diferencia en la hipótesis, la implementación no alcanza un nivel de usabilidad satisfactorio para el usuario. No obstante, no es completamente limitante a los efectos de cumplir con la tarea, sólo la hace más difícil, sobre todo para aquellos usuarios no-bioinformáticos o que manipulen herramientas como línea de comandos. Asumiendo que se espera que estos últimos sean los más abundantes en la plataforma, y que es una tarea requerida, es uno de los primeros aspectos a mejorar. Entre las sugerencias priman aquellas que permitan seleccionar al usuario los datos con los que cuenta. Sin embargo, dado que la información actualmente se obtiene mediante procesos de anotación, una mejora inmediata a trabajar antes que dicha selección es el ingreso de variantes nuevas en un archivo VCF, acompañado de información de la muestra a través de selección en la interfaz. Como producto final, se podría contar con las tres opciones: subir un VCF, una tabla, o simplemente brindar los datos de la variante para su posterior ampliación de características.

También fue explorada la evaluación de pacientes a través de la plataforma. En esta evaluación no pudieron participar todos los usuarios dada la restricción a datos por permisos, por lo que solo pudieron brindar su opinión los usuarios avanzado y competente. Insistiendo en que la plataforma no pretende ser un espacio de interpretación para diagnostico (el cual requeriría de otras características y prestaciones), se buscó evaluar si el caso de uso que se generó como un efecto secundario tuvo éxito en cuanto a su usabilidad. La valoración cualitativa indica que la tarea es factible de realizarse en la plataforma de forma correcta, y con una posible mejora en la experiencia en general, lo que se considera que está principalmente asociada a la visualización y categorización de la información. La tarea de diagnóstico de un paciente, aún así, tiene aspectos que dificultan el proceso, dado que por ejemplo, si bien se puede acceder a las variantes en una lista, no se puede seleccionar las ya evaluadas, o las candidatas (en una planilla, por ejemplo, se podría indicar). Además, es un proceso que está estrictamente asociado a la participación de un administrador para su acceso (dada la privacidad de los datos) y a la participación de una persona que procese los datos previamente de acuerdo a las necesidades de quien se encarga del diagnóstico. En esta instancia no se realizó, pero a futuro, en caso de que se mantenga la aplicación a este tipo de evaluaciones, sería importante contar con medidas cuantitativas, como la comparación de tiempo de evaluación entre la plataforma y otro mecanismo. En este caso, solo se solicitó una medida cualitativa.

La plataforma como un espacio de etiquetado manual también fue evaluada positivamente en rasgos generales. En este caso, se buscó identificar si los usuarios consideran que el sistema generado es acorde al etiquetado de variantes, tanto desde el punto de vista de la posibilidad de etiquetas a generar y su posterior aplicación, como desde los datos que se utilizan para la misma (asumiendo que los insumos con los que se cuenta para generar la etiqueta ya fueron evaluados anteriormente). Un

análisis profundo de las etiquetas generadas, su consenso y la calidad de los datos queda pendiente a la generación de un conjunto mínimo de etiquetas tal que permita su incorporación a los algoritmos generados. Sin embargo, desde una perspectiva de usuario, se ve favorable, más allá de que es preferible que los datos cuenten con la mayor información posible para permitir un veredicto final.

Finalmente se evaluó desde la perspectiva del usuario la función de la plataforma como un espacio de aprendizaje. Atendiendo a que, como se mencionó anteriormente, la plataforma cuenta con pocas herramientas que provengan de un contexto de herramientas de aprendizaje, los usuarios lo valoraron positivamente también. Es importante destacar las diferencias en las evaluaciones y los comentarios que surgen entre el usuario con un nivel principiante del resto. En este sentido se puede observar cómo la el usuario principiante acompaña las afirmaciones, aún así sus requerimientos están más destinados a comprender la tarea de base. Esto sustenta lo ya discutido respecto a que es una plataforma que en su versión inicial cuenta con las funcionalidades mínimas, pero que no alcanza una versión final y completa. El usuario que recién comienza en la tarea de clasificar variantes, y que no sabe de qué se trata, requiere de más ayuda, la cual en la versión actual no se brinda. Sin embargo, los usuarios que ya saben de qué se trata, tienen una mayor soltura y menos exigencias para conseguirlo. Los comentarios de sugerencias recibidos van en sintonía con los planteos discutidos anteriormente.

La evaluación realizada no contó con un conjunto de usuarios muy grande, y se abarcó únicamente desde un punto de vista cualitativo. Teniendo en cuenta las posibilidades de expansión y mejora del sistema, a futuro sería importante contar con herramientas y estrategias para sistematizar la evaluación del uso de la plataforma, por ejemplo, como las brindadas por Google Analytics [274]. Además, se podrían incorporar mecanismos de adquisición de *feedback* de forma regular.

Prueba	Pregunta/ afirmación	Evaluador				
		Experto asesor	**Avanzado externo**	**Avanzado asesor**	**Competente**	**Principiante**
1	La información presentada es suficiente para la tarea de clasificación	4	4	5	5	-
	Hay información de relevancia que no se encuentra y sería importante incluir en versiones posteriores	2	5	5	4	-
	Indicar la información que sería de interés incorporar	*Información específica de ciertas variantes (predictores para frameshift, NMD en stop prematuro, posible afectación de splicing en intrónicas o sinónimas)*	*Criterios a analizar para calificar una variante, agregación familiar (cuando corresponda), datos funcionales*	*Read depth, cigosidad (cuando corresponda), impacto de stop gain en la proteína, impacto de indel (SIFT indel 2).*	*Información de Haplotipos*	*GWAS, Patrones de expresión, saber si es copia unica*
2	La información se encuentra bien categorizada y agrupada en la interfaz	5	5	5	5	5
	La forma en la que se agrupa la información acompaña el uso de reglas	5	5	5	5	-
	La presentación de los datos en la interfaz mejora la experiencia respecto a los métodos usuales	4	5	5	5	4
3	La forma de ingresar nuevas variantes es acorde, cómoda, y directa para el usuario	3	3	3	2	3
	Indicar una sugerencia de mejora para el ingreso de nuevas variantes	*Ingresar de VCF*	*Ampliar posibilidades de acceder mediante filtros*	*Mediante un VCF*	*VCF o interfaz para completar*	*Seleccionar las columnas que corresponden a cada ítem*
4	Se logra evaluar las variantes de un paciente sin pérdida efectiva de información	4	3	5	-	
	Se logra evaluar las variantes de un paciente con una mejora en su experiencia general	3	5	5	-	
	La evaluación podría hacerse más rápido que antes	3	5	5	-	
5	Los insumos finales brindados como etiquetas son acordes para la tarea	5	4	5	5	-
	Las variantes que se ofrecen para brindar etiquetas son útiles	4	5	5	5	-
6	Es una ayuda para el/la principiante	4	5	5	5	4
	Se considera útil en términos de aprendizaje	4	4	5	5	4
	Indicar qué podría complementar una mejor experiencia de aprendizaje	*Veo positivo que se podría hacer una "gamificación"(con puntaje) de la etapa de aprendizaje (y que el sistema da un feedback adecuado para progresar en el aprendizaje). Tal vez una limitante sea (al menos para los médicos) clasificar variantes en abstracto (en oposición a hacerlas para un paciente concreto y real).*	*Incorporación de tutoriales, información de los parámetros que deben evaluarse*	*Completar reglas manualmente*	*Agregar más indicadores de motivación*	*Incorporación de tutoriales, ayudas en cada paso, FAQs*

Cuadro C.1: Resumen de las respuestas adquiridas en los cuestionarios realizados a los usuarios. Las preguntas buscan evaluar 6 aspectos, los cuales están asociados a los experimentos detallados en la Sección C.2. Las respuestas numéricas hacen referencia a una escala del tipo Lickert donde los valores van del 1 al 5, desde "*muy en desacuerdo*" a "*muy de acuerdo*".

Bibliografía

[1] Christian Gilissen et al. «Genome sequencing identifies major causes of severe intellectual disability». En: *Nature* 511.7509 (2014), págs. 344-347.

[2] Dimitri J Stavropoulos et al. «Whole-genome sequencing expands diagnostic utility and improves clinical management in paediatric medicine». En: *NPJ genomic medicine* 1.1 (2016), págs. 1-9.

[3] *Research Area: Health - Rare Diseases*. URL: https://research-and-innovation.ec.europa.eu/research-area/health/rare-diseases_en (visitado 17-09-2020).

[4] Qigang Li et al. «Xrare: a machine learning method jointly modeling phenotypes and genetic evidence for rare disease diagnosis». En: *Genetics in Medicine* 21.9 (2019), págs. 2126-2134.

[5] Karen Eilbeck, Aaron Quinlan y Mark Yandell. «Settling the score: variant prioritization and Mendelian disease». En: *Nature Reviews Genetics* 18.10 (2017), págs. 599-612.

[6] Laura M Amendola et al. «Performance of ACMG-AMP variant-interpretation guidelines among nine laboratories in the Clinical Sequencing Exploratory Research Consortium». En: *The American Journal of Human Genetics* 98.6 (2016), págs. 1067-1076.

[7] Chengliang Dong et al. «Comparison and integration of deleteriousness prediction methods for nonsynonymous SNVs in whole exome sequencing studies». En: *Human molecular genetics* 24.8 (2015), págs. 2125-2137.

[8] Zornitza Stark et al. «Integrating genomics into healthcare: a global responsibility». En: *The American Journal of Human Genetics* 104.1 (2019), págs. 13-20.

[9] Melanie G Pepin et al. «The challenge of comprehensive and consistent sequence variant interpretation between clinical laboratories». En: *Genetics in Medicine* 18.1 (2016), págs. 20-24.

[10] Paris J Vail et al. «Comparison of locus-specific databases for BRCA1 and BRCA2 variants reveals disparity in variant classification within and among databases». En: *Journal of community genetics* 6.4 (2015), págs. 351-359.

[11] S Harrison et al. «Clinical laboratories implement the ACMG/AMP guidelines to resolve differences in variant interpretations submitted to ClinVar». En: *ACMG Annual Clinical Genetics Meeting. Tampa, Florida*. 2016.

[12] H Duzkale et al. «A systematic approach to assessing the clinical significance of genetic variants». En: *Clinical genetics* 84.5 (2013), págs. 453-463.

[13] . *URUGENOMES*. 2020. URL: http://urugenomes.org/.

[14] Lucía Spangenberg et al. «Novel frameshift mutation in PURA gene causes severe encephalopathy of unclear cause». En: *Molecular Genetics & Genomic Medicine* 9.5 (2021), e1622.

[15] Víctor Raggio et al. «Whole genome sequencing reveals a frameshift mutation and a large deletion in YY1AP1 in a girl with a panvascular artery disease». En: *Human Genomics* 15.1 (2021), págs. 1-9.

[16] Camila Simoes et al. «Novel frameshift mutation in LIS1 gene is a probable cause of lissencephaly: a case report». En: *BMC pediatrics* 22.1 (2022), pág. 545.

[17] OrphaNet. *About Rare Diseases*. https://www.orpha.net/consor/cgi-bin/Education_AboutRareDiseases.php?lng=EN&stapage=ST_EDUCATION_EDUCATION_ABOUTRAREDISEASES.

[18] *About Rare Diseases*. 2020. URL: https://rarediseases.info.nih.gov/about.

[19] Stéphanie Nguengang Wakap et al. «Estimating cumulative point prevalence of rare diseases: analysis of the Orphanet database». En: *European Journal of Human Genetics* 28.2 (2020), págs. 165-173.

[20] Caroline F Wright, David R FitzPatrick y Helen V Firth. «Paediatric genomics: diagnosing rare disease in children». En: *Nature Reviews Genetics* 19.5 (2018), págs. 253-268.

[21] Irene Villalón-García et al. «Precision medicine in rare diseases». En: *Diseases* 8.4 (2020), pág. 42.

[22] Alicia Bauskis et al. «The diagnostic odyssey: insights from parents of children living with an undiagnosed condition». En: *Orphanet Journal of Rare Diseases* 17.1 (2022), págs. 1-13.

[23] Lorenza Garrino et al. «Living with and treating rare diseases: experiences of patients and professional health care providers». En: *Qualitative Health Research* 25.5 (2015), págs. 636-651.

[24] Ana Rita Sequeira et al. «The economic and health impact of rare diseases: a meta-analysis». En: *Health Policy and Technology* 10.1 (2021), págs. 32-44.

[25] Nathan Yates y Jennifer Hinkel. «The economics of moonshots: Value in rare disease drug development». En: *Clinical and Translational Science* 15.4 (2022), pág. 809.

[26] Jennifer Baumbusch, Samara Mayer e Isabel Sloan-Yip. «Alone in a crowd? Parents of children with rare diseases' experiences of navigating the healthcare system». En: *Journal of Genetic Counseling* 28.1 (2019), págs. 80-90.

[27] Tiziana Vaisitti et al. «The frequency of rare and monogenic diseases in pediatric organ transplant recipients in Italy». En: *Orphanet journal of rare diseases* 16 (2021), págs. 1-17.

[28] Shruti Marwaha, Joshua W Knowles y Euan A Ashley. «A guide for the diagnosis of rare and undiagnosed disease: beyond the exome». En: *Genome medicine* 14.1 (2022), págs. 1-22.

[29] Joanna S Amberger et al. «OMIM. org: leveraging knowledge across phenotype–gene relationships». En: *Nucleic acids research* 47.D1 (2019), págs. D1038-D1043.

[30] Maria Jackson et al. «The genetic basis of disease». En: *Essays in biochemistry* 62.5 (2018), págs. 643-723.

[31] Mira B Irons y Bruce R Korf. *Human genetics and genomics*. John Wiley & Sons, 2012.

[32] Raymund AC Roos. «Huntington's disease: a clinical review». En: *Orphanet journal of rare diseases* 5 (2010), págs. 1-8.

[33] Virginia C Williams et al. «Neurofibromatosis type 1 revisited». En: *Pediatrics* 123.1 (2009), págs. 124-133.

[34] Michael R Knowles y Peter R Durie. *What is cystic fibrosis?* 2002.

[35] SR Sahoo. «Sickle Cell Anemia-A Brief Synopsis». En: *J Genet Syndr Gene Ther* 11 (2020), pág. 330.

[36] Giancarlo Castaman y Davide Matino. «Hemophilia A and B: molecular and clinical similarities and differences». En: *Haematologica* 104.9 (2019), pág. 1702.

[37] N Gordon. «Colour blindness». En: *Public health* 112.2 (1998), págs. 81-84.

[38] H Moser. «Duchenne muscular dystrophy: pathogenetic aspects and genetic prevention». En: *Human genetics* 66 (1984), págs. 17-40.

[39] Nicky Sirianni et al. «Rett syndrome: confirmation of X-linked dominant inheritance, and localization of the gene to Xq28». En: *The American Journal of Human Genetics* 63.5 (1998), págs. 1552-1557.

[40] Rebecca Dunphy. «Fragile X syndrome». En: *InnovAiT* 13.12 (2020), págs. 712-716.

[41] National Human Genome Research Institute. *Mendelian Inheritance*. URL: https://www.genome.gov/genetics-glossary/Mendelian-Inheritance (visitado 17-09-2020).

[42] Todd J Treangen y Steven L Salzberg. «Repetitive DNA and next-generation sequencing: computational challenges and solutions». En: *Nature Reviews Genetics* 13.1 (2012), págs. 36-46.

[43] Nicole Tabarini et al. «Exploration of Tools for the Interpretation of Human Non-Coding Variants». En: *International Journal of Molecular Sciences* 23.21 (2022), pág. 12977.

[44] Sue Richards 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». En: *Genetics in medicine* 17.5 (2015), págs. 405-423.

[45] Evan E Eichler. «Genetic variation, comparative genomics, and the diagnosis of disease». En: *New England Journal of Medicine* 381.1 (2019), págs. 64-74.

[46] Tuuli Lappalainen et al. «Genomic analysis in the age of human genome sequencing». En: *Cell* 177.1 (2019), págs. 70-84.

[47] Jie Sun et al. «Functional analysis of a nonstop mutation in MITF gene identified in a patient with Waardenburg syndrome type 2». En: *Journal of Human Genetics* 62.7 (2017), págs. 703-709.

[48] Wikipedia. *Synonymous Substitution*. 2020. URL: https://en.wikipedia.org/wiki/Synonymous_substitution.

[49] Darrell L Dinwiddie et al. «De novo frameshift mutation in ASXL3 in a patient with global developmental delay, microcephaly, and craniofacial anomalies». En: *BMC medical genomics* 6 (2013), págs. 1-6.

[50] Hideyuki Nakazawa et al. «A novel germline GATA2 frameshift mutation with a premature stop codon in a family with congenital sensory hearing loss and myelodysplastic syndrome». En: *International Journal of Hematology* 114 (2021), págs. 286-291.

[51] Saverio Brogna y Jikai Wen. «Nonsense-mediated mRNA decay (NMD) mechanisms». En: *Nature structural & molecular biology* 16.2 (2009), págs. 107-113.

[52] *Frameshift Mutation*. URL: https://www.genome.gov/genetics-glossary/Frameshift-Mutation.

[53] Simona Panni et al. «Non-coding RNA regulatory networks». En: *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms* 1863.6 (2020), pág. 194417.

[54] Vittorio Sartorelli y Shannon M Lauberth. «Enhancer RNAs are an important regulatory layer of the epigenome». En: *Nature structural & molecular biology* 27.6 (2020), págs. 521-528.

[55] Elizabeth Santana dos Santos et al. «Non-coding variants in BRCA1 and BRCA2 genes: potential impact on breast and ovarian cancer predisposition». En: *Cancers* 10.11 (2018), pág. 453.

[56] Yury A. Barbitoff et al. «Systematic benchmark of state-of-the-art variant calling pipelines identifies major factors affecting accuracy of coding sequence variant discovery». En: *BMC genomics* 23.1 (2022), pág. 155.

[57] Kym M Boycott et al. «International cooperation to enable the diagnosis of all rare genetic diseases». En: *The American Journal of Human Genetics* 100.5 (2017), págs. 695-705.

[58] Jay Shendure, Gregory M Findlay y Matthew W Snyder. «Genomic medicine–progress, pitfalls, and promise». En: *Cell* 177.1 (2019), págs. 45-57.

[59] Sandhya Verma y Rajesh Kumar Gazara. «Next-generation sequencing: an expedition from workstation to clinical applications». En: *Translational Bioinformatics in Healthcare and Medicine*. Academic Press, 2021, págs. 29-47.

[60] Daniel C. Koboldt. «Best practices for variant calling in clinical sequencing». En: *Genome Medicine* 12.1 (2020), págs. 1-13.

[61] Geraldine A. Van der Auwera et al. «From FastQ data to high-confidence variant calls: the genome analysis toolkit best practices pipeline». En: *Current protocols in bioinformatics* 43.1 (2013), págs. 11-10.

[62] Heng Li. «Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM». En: *arXiv preprint arXiv:1303.3997* (2013).

[63] Heng Li y Richard Durbin. «Fast and accurate short read alignment with Burrows–Wheeler transform». En: *Bioinformatics* 25.14 (2009), págs. 1754-1760.

[64] Heng Li et al. «The sequence alignment/map format and SAMtools». En: *Bioinformatics* 25.16 (2009), págs. 2078-2079.

[65] International Human Genome Sequencing Consortium. «Finishing the euchromatic sequence of the human genome». En: *Nature* 431.7011 (2004), págs. 931-945.

[66] NCBI. *Genome Reference Consortium*. 2020. URL: https://www.ncbi.nlm.nih.gov/grc/help/.

[67] *GATK Best Practices Workflow*. URL: https://gatk.broadinstitute.org/hc/en-us/articles/360035890951 (visitado 17-09-2020).

[68] Broad Institute. *HaplotypeCaller*. 2020. URL: https://gatk.broadinstitute.org/hc/en-us/articles/360037225632-HaplotypeCaller.

[69] Geraldine A. Van der Auwera y Brian D. O'Connor. *Genomics in the Cloud: Using Docker, GATK, and WDL in Terra*. O'Reilly Media, 2020.

[70] Ryan Poplin y et al. «Scaling accurate genetic variant discovery to tens of thousands of samples». En: *BioRxiv* (2017), pág. 201178.

[71] Petr Danecek et al. «The variant call format and VCFtools». En: *Bioinformatics* 27.15 (2011), págs. 2156-2158.

[72] Natasha T Strande et al. «Navigating the nuances of clinical sequence variant interpretation in Mendelian disease». En: *Genetics in Medicine* 20.9 (2018), págs. 918-926.

[73] 1000 Genomes Project Consortium et al. «A global reference for human genetic variation». En: *Nature* 526.7571 (2015), pág. 68.

[74] Wenqing Fu et al. «Analysis of 6,515 exomes reveals the recent origin of most human protein-coding variants». En: *Nature* 493.7431 (2013), págs. 216-220.

[75] University of Washington. *Environmental and Occupational Health Sciences*. 2020. URL: https://evs.gs.washington.edu/EVS/.

[76] Konrad J Karczewski et al. «The ExAC browser: displaying reference data information from over 60 000 exomes». En: *Nucleic acids research* 45.D1 (2017), págs. D840-D845.

[77] CSER Consortium. *The Clinical Sequencing Exploratory Research (CSER) Consortium*. 2020. URL: https://cser1.cser-consortium.org/.

[78] Keith Nykamp et al. «Sherloc: a comprehensive refinement of the ACMG–AMP variant classification criteria». En: *Genetics in Medicine* 19.10 (2017), págs. 1105-1117.

[79] Izabela Karbassi et al. «A standardized DNA variant scoring system for pathogenicity assessments in Mendelian disorders». En: *Human mutation* 37.1 (2016), págs. 127-134.

[80] Heidi L Rehm et al. «ClinGen—the clinical genome resource». En: *New England Journal of Medicine* 372.23 (2015), págs. 2235-2242.

[81] Ahmad N Abou Tayoun et al. «Recommendations for interpreting the loss of function PVS1 ACMG/AMP variant criterion». En: *Human mutation* 39.11 (2018), págs. 1517-1524.

[82] Leslie G Biesecker y Steven M Harrison. «The ACMG/AMP reputable source criteria for the interpretation of sequence variants». En: *Genetics in Medicine* 20.12 (2018), págs. 1687-1688.

[83] Sarah E Brnich et al. «Recommendations for application of the functional evidence PS3/BS3 criterion using the ACMG/AMP sequence variant interpretation framework». En: *Genome medicine* 12.1 (2020), págs. 1-12.

[84] Xi Luo et al. «ClinGen myeloid malignancy variant curation expert panel recommendations for germline RUNX1 variants». En: *Blood advances* 3.20 (2019), págs. 2962-2979.

[85] *Wintervar.wglab.org*. URL: http://wintervar.wglab.org/.

[86] Quan Li y Kai Wang. «InterVar: clinical interpretation of genetic variants by the 2015 ACMG-AMP guidelines». En: *The American Journal of Human Genetics* 100.2 (2017), págs. 267-280.

[87] Ronak Y Patel et al. «ClinGen Pathogenicity Calculator: a configurable system for assessing pathogenicity of genetic variants». En: *Genome Medicine* 9.1 (2017), págs. 1-9.

[88] Christine G Preston et al. «ClinGen Variant Curation Interface: a variant classification platform for the application of evidence criteria from ACMG/AMP guidelines». En: *Genome Medicine* 14.1 (2022), pág. 6.

[89] Christos Kopanos et al. «VarSome: the human genomic variant search engine». En: *Bioinformatics* 35.11 (2019), pág. 1978.

[90] *Varsome*. URL: https://varsome.com/.

[91] *Franklin*. URL: https://franklin.genoox.com/ (visitado 17-09-2020).

[92] B Quintáns et al. «Medical genomics: The intricate path from genetic variant identification to clinical interpretation». En: *Applied & translational genomics* 3.3 (2014), págs. 60-67.

[93] Kai Wang, Mingyao Li y Hakon Hakonarson. «ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data». En: *Nucleic acids research* 38.16 (2010), e164-e164.

[94] Pablo Cingolani. «Variant annotation and functional prediction: SnpEff». En: *Variant Calling: Methods and Protocols*. Springer, 2012, págs. 289-314.

[95] William McLaren et al. «The ensembl variant effect predictor». En: *Genome biology* 17.1 (2016), págs. 1-14.

[96] Hao Hu et al. «VAAST 2.0: Improved variant classification and disease-gene identification using a conservation-controlled amino acid substitution matrix». En: *Genetic epidemiology* 37.6 (2013), págs. 622-634.

[97] Sarah B Ng et al. «Targeted capture and massively parallel sequencing of 12 human exomes». En: *Nature* 461.7261 (2009), págs. 272-276.

[98] *myvariant.info*. URL: https://myvariant.info/.

[99] Marc V Singleton et al. «Phevor combines multiple biomedical ontologies for accurate identification of disease-causing alleles in single individuals and small nuclear families». En: *The American Journal of Human Genetics* 94.4 (2014), págs. 599-610.

[100] Damian Smedley et al. «Next-generation diagnostics and disease-gene discovery with the Exomiser». En: *Nature protocols* 10.12 (2015), págs. 2004-2015.

[101] Hui Yang, Peter N Robinson y Kai Wang. «Phenolyzer: phenotype-based prioritization of candidate genes for human diseases». En: *Nature methods* 12.9 (2015), págs. 841-843.

[102] Stanford University. *Amelie*. 2020. URL: https://amelie.stanford.edu/.

[103] Peter D Stenson et al. «The Human Gene Mutation Database (HGMD®): optimizing its use in a clinical diagnostic or research setting». En: *Human genetics* 139 (2020), págs. 1197-1207.

[104] Melissa J Landrum et al. «ClinVar: public archive of interpretations of clinically relevant variants». En: *Nucleic acids research* 44.D1 (2016), págs. D862-D868.

[105] James A Diao, Isaac S Kohane y Arjun K Manrai. «Biomedical informatics and machine learning for clinical genomics». En: *Human molecular genetics* 27.R1 (2018), R29-R34.

[106] Melissa J Landrum et al. «ClinVar: public archive of relationships among sequence variation and human phenotype». En: *Nucleic acids research* 42.D1 (2014), págs. D980-D985.

[107] Konrad Karczewski y LJML Francioli. «The genome aggregation database (gnomAD)». En: *MacArthur Lab* (2017), págs. 1-10.

[108] Eric J Topol. «High-performance medicine: the convergence of human and artificial intelligence». En: *Nature medicine* 25.1 (2019), págs. 44-56.

[109] Raquel Dias y Ali Torkamani. «Artificial intelligence in clinical and genomic diagnostics». En: *Genome medicine* 11.1 (2019), págs. 1-12.

[110] Johannes Birgmeier et al. «AMELIE speeds Mendelian diagnosis by matching patient phenotype and genotype to primary literature». En: *Science Translational Medicine* 12.544 (2020), eaau9113.

[111] Johannes Birgmeier et al. «AVADA: toward automated pathogenic variant evidence retrieval directly from the full-text literature». En: *Genetics in Medicine* 22.2 (2020), págs. 362-370.

[112] Francisco M De La Vega et al. «Artificial intelligence enables comprehensive genome interpretation and nomination of candidate diagnoses for rare genetic diseases». En: *Genome Medicine* 13 (2021), págs. 1-19.

[113] Maxwell W Libbrecht y William Stafford Noble. «Machine learning applications in genetics and genomics». En: *Nature Reviews Genetics* 16.6 (2015), págs. 321-332.

[114] Žiga Avsec. «Kipoi: accelerating the community exchange and reuse of predictive models for genomics». En: *ICML Workshop for Computational Biology*. 2018.

[115] Sarah J MacEachern y Nils D Forkert. «Machine learning for precision medicine». En: *Genome* 64.4 (2021), págs. 416-425.

[116] Barthélémy Caron, Yufei Luo y Antonio Rausell. «NCBoost classifies pathogenic non-coding variants in Mendelian diseases through supervised learning on purifying selection signals in humans». En: *Genome biology* 20 (2019), págs. 1-22.

[117] Damian Smedley et al. «A whole-genome analysis framework for effective identification of pathogenic regulatory variants in Mendelian disease». En: *The American Journal of Human Genetics* 99.3 (2016), págs. 595-606.

[118] Gökcen Eraslan et al. «Deep learning: new computational modelling techniques for genomics». En: *Nature Reviews Genetics* 20.7 (2019), págs. 389-403.

[119] Jian Zhou y Olga G Troyanskaya. «Predicting effects of noncoding variants with deep learning–based sequence model». En: *Nature methods* 12.10 (2015), págs. 931-934.

[120] *Create*. URL: http://deepsea.princeton.edu/job/analysis/create/ (visitado 17-09-2020).

[121] David R Kelley et al. «Sequential regulatory activity prediction across chromosomes with convolutional neural networks». En: *Genome research* 28.5 (2018), págs. 739-750.

[122] Jian Zhou et al. «Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk». En: *Nature genetics* 50.8 (2018), págs. 1171-1179.

[123] Žiga Avsec et al. «Effective gene expression prediction from sequence by integrating long-range interactions». En: *Nature methods* 18.10 (2021), págs. 1196-1203.

[124] Ivan A Adzhubei et al. «A method and server for predicting damaging missense mutations». En: *Nature methods* 7.4 (2010), págs. 248-249.

[125] Daniel Quang, Yifei Chen y Xiaohui Xie. «DANN: a deep learning approach for annotating the pathogenicity of genetic variants». En: *Bioinformatics* 31.5 (2015), págs. 761-763.

[126] Hashem A Shihab et al. «An integrative approach to predicting the functional effects of non-coding and coding sequence variation». En: *Bioinformatics* 31.10 (2015), págs. 1536-1543.

[127] Karthik A Jagadeesh et al. «M-CAP eliminates a majority of variants of uncertain significance in clinical exomes at high sensitivity». En: *Nature genetics* 48.12 (2016), pág. 1581.

[128] Nilah M Ioannidis et al. «REVEL: an ensemble method for predicting the pathogenicity of rare missense variants». En: *The American Journal of Human Genetics* 99.4 (2016), págs. 877-885.

[129] Abel González-Pérez y Nuria López-Bigas. «Improving the assessment of the outcome of nonsynonymous SNVs with a consensus deleteriousness score, Condel». En: *The American Journal of Human Genetics* 88.4 (2011), págs. 440-449.

[130] SungHwan Kim et al. «Meta-analytic support vector machine for integrating multiple omics data». En: *BioData mining* 10 (2017), págs. 1-14.

[131] *CADD*. URL: https://cadd.bihealth.org/.

[132] Philipp Rentzsch et al. «CADD: predicting the deleteriousness of variants throughout the human genome». En: *Nucleic acids research* 47.D1 (2019), págs. D886-D894.

[133] Hashem A Shihab et al. «Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models». En: *Human mutation* 34.1 (2013), págs. 57-65.

[134] Asif Javed, Saloni Agrawal y Pauline C Ng. «Phen-Gen: combining phenotype and genotype to analyze rare disorders». En: *Nature methods* 11.9 (2014), págs. 935-937.

[135] Alejandro Sifrim et al. «eXtasy: variant prioritization by genomic data fusion». En: *Nature methods* 10.11 (2013), págs. 1083-1084.

[136] Peter N Robinson et al. «Improved exome prioritization of disease genes through cross-species phenotype comparison». En: *Genome research* 24.2 (2014), págs. 340-348.

[137] Nicolas Fiorini et al. «How user intelligence is improving PubMed». En: *Nature biotechnology* 36.10 (2018), págs. 937-945.

[138] *National Center for Biotechnology Information PMC*. URL: https://www.ncbi.nlm.nih.gov/pmc/.

[139] Kyubum Lee, Chih-Hsuan Wei y Zhiyong Lu. «Recent advances of automated methods for searching and extracting genomic variant information from biomedical literature». En: *Briefings in bioinformatics* 22.3 (2021), bbaa142.

[140] «Large-scale discovery of novel genetic causes of developmental disorders». En: *Nature* 519.7542 (2015), págs. 223-228.

[141] *Gene Names*. URL: https://www.genenames.org/.

[142] UniProt Consortium. «UniProt: a hub for protein information». En: *Nucleic acids research* 43.D1 (2015), págs. D204-D212.

[143] *HGMD*. URL: https://www.hgmd.cf.ac.uk/.

[144] Michelle M Clark et al. «Diagnosis of genetic diseases in seriously ill children by rapid whole-genome sequencing and automated phenotyping and interpretation». En: *Science translational medicine* 11.489 (2019), eaat6177.

[145] Eduardo Pérez-Palma et al. «Simple ClinVar: an interactive web server to explore and retrieve gene and disease variants aggregated in ClinVar database». En: *Nucleic Acids Research* 47.W1 (2019), W99-W105.

[146] Stephen T Sherry et al. «dbSNP: the NCBI database of genetic variation». En: *Nucleic acids research* 29.1 (2001), págs. 308-311.

[147] David L Wheeler et al. «Database resources of the National Center for Biotechnology Information. Nucleic Acids Res». En: (2005).

[148] Nuala A O'Leary et al. «Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation». En: *Nucleic acids research* 44.D1 (2016), págs. D733-D745.

[149] Steven M Harrison et al. «Using ClinVar as a resource to support variant interpretation». En: *Current protocols in human genetics* 89.1 (2016), págs. 8-16.

[150] *ClinVar Review Status*. URL: https://www.ncbi.nlm.nih.gov/clinvar/docs/review_status/.

[151] *ClinVar*. URL: https://www.ncbi.nlm.nih.gov/clinvar/.

[152] *Perl.org*. URL: https://www.perl.org/.

[153] Wikipedia. *ANNOVAR*. 2020. URL: https://en.wikipedia.org/wiki/ANNOVAR.

[154] Hai Lin et al. «RegSNPs-intron: a computational framework for predicting pathogenic impact of intronic single nucleotide variants». En: *Genome biology* 20 (2019), págs. 1-16.

[155] Xiaoming Liu et al. «dbNSFP v4: a comprehensive database of transcript-specific functional predictions and annotations for human nonsynonymous and splice-site SNVs». En: *Genome medicine* 12.1 (2020), págs. 1-8.

[156] Graham RS Ritchie et al. «Functional annotation of noncoding sequence variants». En: *Nature methods* 11.3 (2014), págs. 294-296.

[157] Gustavo Glusman et al. «Kaviar: an accessible system for testing SNV novelty». En: *Bioinformatics* 27.22 (2011), págs. 3216-3217.

[158] Iuliana Ionita-Laza et al. «A spectral approach integrating functional genomic annotations for coding and noncoding variants». En: *Nature genetics* 48.2 (2016), págs. 214-220.

[159] *Dash documentation & user guide*. URL: https://dash.plotly.com/.

[160] *Flask*. URL: https://flask.palletsprojects.com/en/1.1.x/.

[161] *MySQK*. URL: https://www.mysql.com/.

[162] Winston Chang et al. «shiny: Web application framework for r, 2015». En: *R package version* 1.0 (2018), p25.

[163] *Streamlit-the fastest way to build and share data apps*. URL: https://streamlit.io/.

[164] *Voilà dashboards*. URL: https://github.com/voila-dashboards.

[165] *Panel Documentation*. 2022. URL: https://panel.holoviz.org/.

[166] *Shiny*. URL: https://shiny.rstudio.com/.

[167] Lihua Jia et al. «Development of interactive biological web applications with R/Shiny». En: *Briefings in Bioinformatics* 23.1 (2022), bbab415.

[168] *Repositorio GitLab del proyecto*. URL: https://gitlab.com/ingenieriabiologica/investigacion/maestria_camila_bioinfo.

[169] *Shiny for Python*. URL: https://shiny.rstudio.com/.

[170] Amber N Habowski, TJ Habowski y ML Waterman. «GECO: gene expression clustering optimization app for non-linear data visualization of patterns». En: *BMC bioinformatics* 22.1 (2021), págs. 1-13.

[171] Furkan M Torun et al. «Transparent exploration of machine learning for biomarker discovery from proteomics and omics data». En: *Journal of Proteome Research* ().

[172] Mohammad Khorasani, Mohamed Abdou y Javier Hernández Fernández. «Getting Started with Streamlit». En: *Web Application Development with Streamlit*. Springer, 2022, págs. 1-30.

[173] Data Revenue. *Data Dashboarding: Streamlit vs Dash vs Shiny vs Voila*. 2020. URL: https://www.datarevenue.com/en-blog/data-dashboarding-streamlit-vs-dash-vs-shiny-vs-voila (visitado 17-09-2020).

[174] David Granjon. *Outstanding User Interfaces with Shiny*. CRC Press, 2022.

[175] *Comparing Dash, Shiny, and Streamlit*. URL: https://plotly.com/comparing-dash-shiny-streamlit/ (visitado 17-09-2020).

[176] *Dash Enterprise*. URL: https://dash.plotly.com/dash-enterprise (visitado 17-09-2020).

[177] *JavaScript*. 2020. URL: https://www.javascript.com/.

[178] *React – a JavaScript library for building user interfaces*. URL: https://reactjs.org/.

[179] *Plotly Python*. Accessed: [insert date]. URL: https://plotly.com/python/.

[180] *Plotly JavaScript*. URL: https://plotly.com/javascript/.

[181] Elias Dabbas. *Interactive Dashboards and Data Apps with Plotly and Dash: Harness the power of a fully fledged frontend web framework in Python–no JavaScript required*. Packt Publishing Ltd, 2021.

[182] CNN News. *Plotly Dash or React.js + Plotly.js*. 2020. URL: https://towardsdatascience.com/plotly-dash-or-react-js-plotly-js-b491b3615512.

[183] *Dash Core Components*. URL: https://dash.plotly.com/dash-core-components (visitado 17-09-2020).

[184] *Dash HTML Components*. URL: https://dash.plotly.com/dash-html-components (visitado 17-09-2020).

[185] *Dash Bootstrap Components*. URL: https://dash-bootstrap-components.opensource.faculty.ai/ (visitado 17-09-2020).

[186] W3Schools. *HTML Tutorial*. 2020. URL: https://www.w3schools.com/html/.

[187] W3Schools. *CSS Tutorial*. 2020. URL: https://www.w3schools.com/css/.

[188] *Dash Bio*. URL: https://dash.plotly.com/dash-bio (visitado 17-09-2020).

[189] Shammamah Hossain. «Visualization of bioinformatics data with dash bio». En: 2019.

[190] Edgar F Codd. «A relational model of data for large shared data banks». En: *Communications of the ACM* 13.6 (1970), págs. 377-387.

[191] Wade L Schulz et al. «Evaluation of relational and NoSQL database architectures to manage genomic annotations». En: *Journal of biomedical informatics* 64 (2016), págs. 288-295.

[192] Soarov Chakraborty, Shourav Paul y KM Azharul Hasan. «Performance comparison for data retrieval from nosql and sql databases: a case study for covid-19 genome sequence dataset». En: *2021 2nd International Conference on Robotics, electrical and signal processing techniques (ICREST)*. IEEE. 2021, págs. 324-328.

[193] Michael Stonebraker. «SQL databases v. NoSQL databases». En: *Communications of the ACM* 53.4 (2010), págs. 10-11.

[194] Zachary Parker, Scott Poe y Susan V Vrbsky. «Comparing nosql mongodb to an sql db». En: *Proceedings of the 51st ACM Southeast Conference*. 2013, págs. 1-6.

[195] Shicai Wang et al. «High dimensional biological data retrieval optimization with NoSQL technology». En: *BMC genomics*. Vol. 15. 8. Springer. 2014, págs. 1-8.

[196] Alejandro Brenes et al. «The Encyclopedia of Proteome Dynamics: a big data ecosystem for (prote) omics». En: *Nucleic Acids Research* 46.D1 (2018), págs. D1202-D1209.

[197] Min He et al. «SeqHBase: a big data toolset for family based sequencing data analysis». En: *Journal of medical genetics* 52.4 (2015), págs. 282-288.

[198] Maxim Barenboim y Thomas Manke. «ChroMoS: an integrated web tool for SNP classification, prioritization and functional interpretation». En: *Bioinformatics* 29.17 (2013), págs. 2197-2198.

[199] Umadevi Paila et al. «GEMINI: integrative exploration of genetic variation and genome annotations». En: *PLoS computational biology* 9.7 (2013), e1003153.

[200] Matteo Gabetta et al. «BigQ: a NoSQL based framework to handle genomic variants in i2b2». En: *BMC bioinformatics* 16.1 (2015), págs. 1-11.

[201] Christine G Preston et al. «ClinGen Variant Curation Interface: A Variant Classification Platform for the Application of Evidence Criteria from ACMG/AMP Guidelines». En: *medRxiv* (2021).

[202] MA Bouzinier et al. «AnFiSA: An open-source computational platform for the analysis of sequencing data for rare genetic disease». En: *Journal of Biomedical Informatics* 133 (2022), pág. 104174.

[203] Hufeng Zhou et al. «FAVOR: functional annotation of variants online resource and annotator for variation across the human genome». En: *Nucleic Acids Research* 51.D1 (2023), págs. D1300-D1311.

[204] Subhajit Pal et al. «Big data in biology: The hope and present-day challenges in it». En: *Gene Reports* 21 (2020), pág. 100869.

[205] Fabrizio Celesti et al. «Optimizing the research of dna sequences in a nosql document database: A preliminary study». En: *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE. 2019, págs. 1153-1158.

[206] Xiaoming Wang et al. «Big data management challenges in health research—a literature review». En: *Briefings in bioinformatics* 20.1 (2019), págs. 156-167.

[207] Rodrigo Aniceto et al. «Evaluating the cassandra NoSQL database approach for genomic data persistency». En: *International journal of genomics* 2015 (2015).

[208] Jack Chan, Ray Chung y Jack Huang. *Python API Development Fundamentals: Develop a full-stack web application with Python and Flask*. Packt Publishing Ltd, 2019.

[209] David T Miller et al. «ACMG SF v3. 0 list for reporting of secondary findings in clinical exome and genome sequencing: a policy statement of the American College of Medical Genetics and Genomics (ACMG)». En: *Genetics in medicine* 23.8 (2021), págs. 1381-1390.

[210] Karen Eilbeck et al. «The Sequence Ontology: a tool for the unification of genome annotations». En: *Genome biology* 6.5 (2005), págs. 1-12.

[211] *Online Mendelian Inheritance in Man, OMIM®*. 2022. URL: https://omim.org/.

[212] *NCBI MedGen*. URL: https://www.ncbi.nlm.nih.gov/medgen.

[213] Russell Romney. *Dash Auth Flow*. URL: https://github.com/russellromney/dash-auth-flow.

[214] Rafael Miquelino. *dash-flask-login*. 2020. URL: https://github.com/RafaelMiquelino/dash-flask-login.

[215] SQLite. *SQLite Home Page*. https://www.sqlite.org/index.html. 2020.

[216] *Layout*. 2020. URL: https://dash.plotly.com/layout.

[217] Plotly. *Basic Callbacks*. 2020. URL: https://dash.plotly.com/basic-callbacks.

[218] *ClinVar*. URL: https://ftp.ncbi.nlm.nih.gov/pub/clinvar/.

[219] *MySQLClient*. URL: https://mysqlclient.readthedocs.io/.

[220] *SimpleJSON*. 2020. URL: https://simplejson.readthedocs.io/en/latest/.

[221] *Swagger.io*. URL: https://swagger.io/.

[222] *API Bioinformatics Documentation*. URL: http://127.0.0.1:5000/api/v1/doc/#/.

[223] *HTTP Methods*. 2020. URL: https://www.w3schools.com/tags/ref_httpmethods.asp.

[224] Behzad Tabibian et al. «Enhancing human learning via spaced repetition optimization». En: *Proceedings of the National Academy of Sciences* 116.10 (2019), págs. 3988-3993.

[225] Burr Settles y Brendan Meeder. «A trainable spaced repetition model for language learning». En: *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*. 2016, págs. 1848-1858.

[226] P Wozniak. *Application of a computer to improve the results obtained in working with the SuperMemo method. 1990*. 2016.

[227] *supermemo2*. 2020. URL: https://pypi.org/project/supermemo2/.

[228] Dimitar Goshevski, Joana Veljanoska y Thanos Hatziapostolou. «A review of gamification platforms for higher education». En: *Proceedings of the 8th Balkan Conference in Informatics*. 2017, págs. 1-6.

[229] Shawn Loewen et al. «Mobile-assisted language learning: A Duolingo case study». En: *ReCALL* 31.3 (2019), págs. 293-311.

[230] Siobhan O'Donovan, James Gain y Patrick Marais. «A case study in the gamification of a university-level games development course». En: *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*. 2013, págs. 242-251.

[231] Hui Wen Goh, Ulyana Tkachenko y Jonas Mueller. «Utilizing supervised models to infer consensus labels and their quality from data with multiple annotators». En: *arXiv preprint arXiv:2210.06812* (2022).

[232] Dmitry Ustalov et al. «A General-Purpose Crowdsourcing Computational Quality Control Toolkit for Python». Inglés. En: *The Ninth AAAI Conference on Human Computation and Crowdsourcing: Works-in-Progress and Demonstration Track*. HCOMP 2021. 2021. arXiv: 2109.08584 [cs.HC]. URL: https://www.humancomputation.com/2021/assets/wips_demos/HCOMP_2021_paper_85.pdf.

[233] *actG-Learn*. URL: http://munch.paap.cup.edu.uy:8050/ (visitado 17-09-2020).

[234] *Repositorio GitLab del proyecto*. URL: https://gitlab.com/ingenieriabiologica/investigacion/maestria_camila_bioinfo/-/tree/master/dash/dash-auth-flow.

[235] *Gunicorn*. URL: https://gunicorn.org/.

[236] Camila Simoes. *Prueba del funcionamiento general de actG-Learn*. URL: https://youtu.be/Rn0Ok4FQ91o.

[237] Plotly Dash. *The Main Performance Limitation of Plotly Dash and How to Fix It*. 2020. URL: https://dash.plotly.com/performance#:~:text=The%20main%20performance%20limitation%20of,your%20app%20will%20feel%20snappier..

[238] Benoît Choffin et al. «DAS3H: modeling student learning and forgetting for optimally scheduling distributed practice of skills». En: *arXiv preprint arXiv:1905.06873* (2019).

[239] Sarah Alaghbari et al. «A User-Centered Approach to Gamify the Manual Creation of Training Data for Machine Learning». En: *i-com* 20.1 (2021), págs. 33-48.

[240] Natasa Gisev, J Simon Bell y Timothy F Chen. «Interrater agreement and interrater reliability: key concepts, approaches, and applications». En: *Research in Social and Administrative Pharmacy* 9.3 (2013), págs. 330-338.

[241] Janyce Wiebe, Rebecca Bruce y Thomas P O'Hara. «Development and use of a gold-standard data set for subjectivity classifications». En: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*. 1999, págs. 246-253.

[242] Jerónimo Hernández-González, Iñaki Inza y Jose A Lozano. «Multidimensional learning from crowds: Usefulness and application of expertise detection». En: *International Journal of Intelligent Systems* 30.3 (2015), págs. 326-354.

[243] James Zou et al. «A primer on deep learning in genomics». En: *Nature genetics* 51.1 (2019), págs. 12-18.

[244] Ye Liu et al. «Computational approaches for predicting variant impact: An overview from resources, principles to applications». En: *Frontiers in Genetics* 13 (2022).

[245] Richard Grantham. «Amino acid difference formula to help explain protein evolution». En: *science* 185.4154 (1974), págs. 862-864.

[246] Pauline C Ng y Steven Henikoff. «SIFT: Predicting amino acid changes that affect protein function». En: *Nucleic acids research* 31.13 (2003), págs. 3812-3814.

[247] Sung Chun y Justin C Fay. «Identification of deleterious mutations within three human genomes». En: *Genome research* 19.9 (2009), págs. 1553-1561.

[248] Jana Marie Schwarz et al. «MutationTaster evaluates disease-causing potential of sequence alterations». En: *Nature methods* 7.8 (2010), págs. 575-576.

[249] Boris Reva, Yevgeniy Antipin y Chris Sander. «Predicting the functional impact of protein mutations: application to cancer genomics». En: *Nucleic acids research* 39.17 (2011), e118-e118.

[250] Yongwook Choi et al. «Predicting the functional effect of amino acid substitutions and indels». En: *PloS one* 7.10 (2012), e46688.

[251] Hannah Carter et al. «Identifying Mendelian disease genes with the variant effect scoring tool». En: *BMC genomics* 14.3 (2013), págs. 1-16.

[252] Brad Gulko et al. «Probabilities of fitness consequences for point mutations across the human genome». En: *bioRxiv* (2014), pág. 006825.

[253] Gregory M Cooper et al. «Single-nucleotide evolutionary constraint scores highlight disease-causing mutations». En: *Nature methods* 7.4 (2010), págs. 250-251.

[254] Katherine S Pollard et al. «Detection of nonneutral substitution rates on mammalian phylogenies». En: *Genome research* 20.1 (2010), págs. 110-121.

[255] Joseph Felsenstein y Gary A Churchill. «A Hidden Markov Model approach to variation among sites in rate of evolution.» En: *Molecular biology and evolution* 13.1 (1996), págs. 93-104.

[256] Kerstin Lindblad-Toh et al. «A high-resolution map of human evolutionary constraint using 29 mammals». En: *Nature* 478.7370 (2011), págs. 476-482.

[257] Stef Van Buuren. *Flexible imputation of missing data*. CRC press, 2018.

[258] Jerome H Friedman. «Greedy function approximation: a gradient boosting machine». En: *Annals of statistics* (2001), págs. 1189-1232.

[259] Guolin Ke et al. «Lightgbm: A highly efficient gradient boosting decision tree». En: *Advances in neural information processing systems* 30 (2017).

[260] Ben Omega Petrazzini et al. «Evaluation of different approaches for missing data imputation on features associated to genomic data». En: *BioData mining* 14.1 (2021), págs. 1-13.

[261] scikit-learn. *KNeighborsClassifier*. Accessed: [insert date]. 2020. URL: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html.

[262] scikit-learn. *Gaussian Naive Bayes (GaussianNB)*. Accessed: [insert date]. 2020. URL: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html.

[263] scikit-learn. *SVC*. 2020. URL: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn-svm-svc.

[264] scikit-learn. *DecisionTreeClassifier*. 2020. URL: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.

[265] scikit-learn. *RandomForestClassifier*. Accessed: [insert date]. 2020. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[266] *XGBoost Python API*. [Online; accessed ¡today¿]. URL: https://xgboost.readthedocs.io/en/stable/python/python_api.html.

[267] Aurelien Geron. «Handson Machine Learning with Scikitlearn, Keras & TensorFlow. o'Reiley Media». En: *Inc, Sebatopol, CA* (2019).

[268] scikit-learn. *Cross-validation: evaluating estimator performance*. 2020. URL: https://scikit-learn.org/stable/modules/cross_validation.html.

[269] Scikit-Learn. *Exhaustive Grid Search*. 2020. URL: https://scikit-learn.org/stable/modules/grid_search.html#exhaustive-grid-search.

[270] Fabian Pedregosa et al. «Scikit-learn: Machine learning in Python». En: *the Journal of machine Learning research* 12 (2011), págs. 2825-2830.

[271] Hamed Valizadegan, Quang Nguyen y Milos Hauskrecht. «Learning classification models from multiple experts». En: *Journal of biomedical informatics* 46.6 (2013), págs. 1125-1135.

[272] SAMTools. *VCFv4.2*. URL: https://samtools.github.io/hts-specs/VCFv4.2.pdf (visitado 17-09-2020).

[273] Ankur Joshi et al. «Likert scale: Explored and explained». En: *British journal of applied science & technology* 7.4 (2015), pág. 396.

[274] *Google Analytics*. Accessed: 2020-09-14. URL: https://analytics.google.com/analytics/web/#/report-home/a80128377w119337398p124874219.