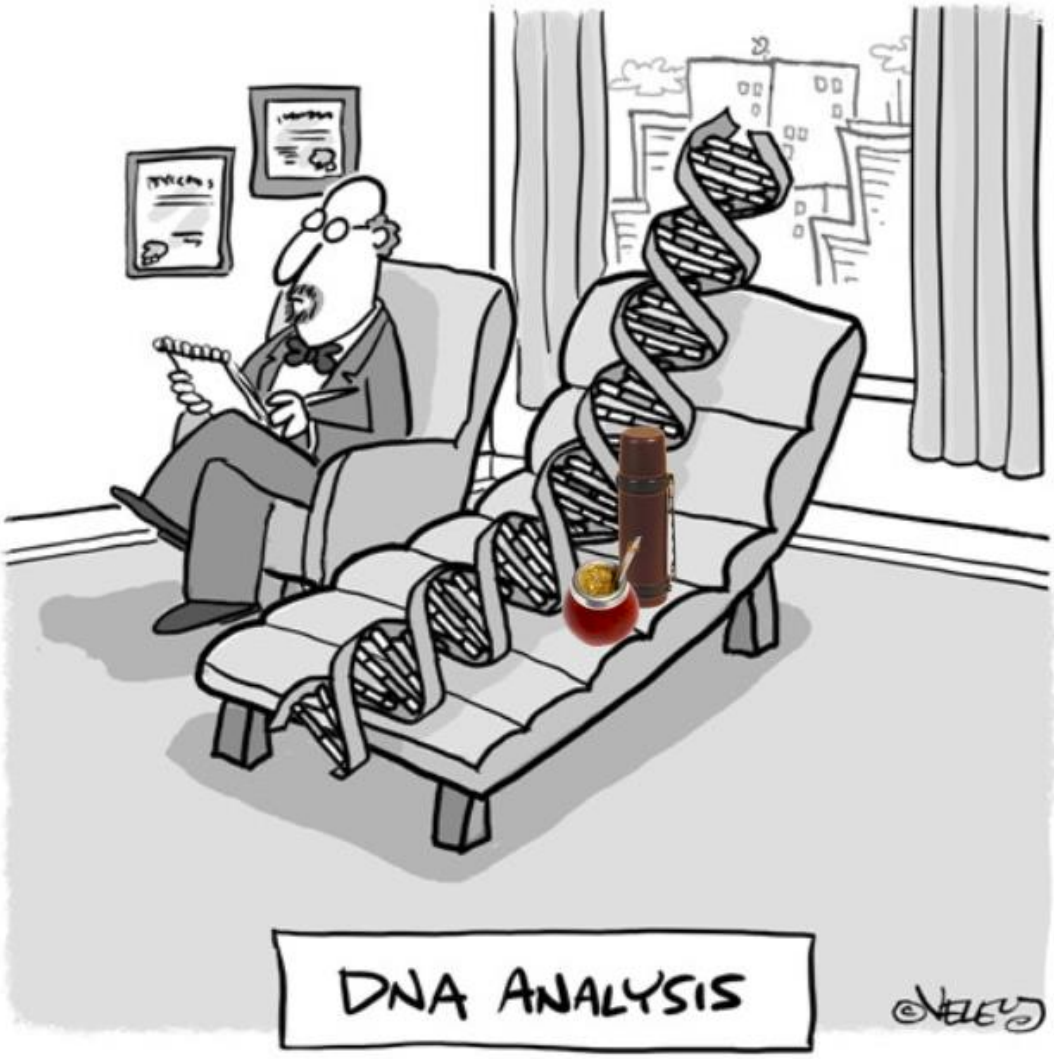
Identificando variantes patogénicas en regiones no codificantes del genoma humano

Ben Omega Petrazzini

Tutora: Lucia Spangenberg

Co-tutor: Fernando López-Bello



Introducción

Enfermedades raras

- Definición
- Relevancia
- Características

Secuenciación masiva

- Surgimiento
- Aplicaciones bioinformáticas a datos de secuenciación masiva
- Secuenciación masiva para el estudio de regiones no-codificantes

Aprendizaje Automático

- Definición
- Aplicaciones a bases de datos biológicas

Proyecto URUGENOMES

Metodología

Esquema de trabajo

Obtención de datos de la base de datos ClinVar

- Versión utilizada
- Funcionamiento de la base y como respalda su anotación
- Etiquetado de la Significancia Clínica de las variantes

Anotación de características biológicas

- Funcionamiento de ANNOVAR
- Descripción de las bases de datos interrogadas para la anotación

Evaluación de algoritmos de inferencia e imputación de valores ausentes en las anotaciones

- Descripción de los algoritmos
- Esquema de trabajo para la evaluación de los algoritmos
- Técnica alternativa para la imputación de *regsnp*

Pre-procesamiento de datos

- Correlación entre variables

Algoritmos utilizados

- Descripción de los algoritmos de Aprendizaje Automático

Definición de las particiones evaluadas y entrenamiento de los modelos

- Descripción de las tres particiones evaluadas
- Paquetes utilizados para entrenar los modelos

Selección de hiperparámetros, evaluación y comparación de los modelos

- Validación cruzada para seleccionar hiperparámetros
- Matrices de confusión y curvas ROC

Implementación en genomas uruguayos

- Traslado del esquema de trabajo a datos genómicos de pacientes

Resultados

Desempeño comparado de algoritmos para inferir valores ausentes

- Desempeño de cada algoritmo por columna
- Desempeño del Perceptrón Multicapa en predecir valores ausentes en *regsnp*

Correlación de variables anotadas

- Variables descartadas

Selección de hiperparámetros y validación interna de cada algoritmo

- Desempeño del escenario SI-SC
- Desempeño del escenario TC
- Desempeño del escenario B-P

Evaluación de cada algoritmo en set de datos externo del escenario B-P

- Matrices de confusión y curvas ROC
- Evaluación del sobreajuste
- Confianza de la predicción por clase de los modelos GBT y RF

Priorización de variantes en genomas uruguayos

Discusión

Imputación de valores ausentes

- Algoritmos basados en vecinos más cercanos para la inferencia de valores ausentes en datos genómicos
- La imputación varía según la naturaleza biológica de los datos inferidos

Estudio de escenarios

- Distribución binaria más eficiente para la predicción de significancia clínica en variantes humanas

Algoritmos de buen desempeño en evaluación

- Modelos basados en árboles para la predicción de significancia clínica en variantes no codificantes
- Evaluación e implicancia del sobreajuste de los modelos

Variantes priorizadas en genomas uruguayos

Agradecimientos

Anexo

Introducción

Definición de enfermedades raras.

Clasificar una patología como enfermedad rara (ER) no es una tarea trivial, existen dos definiciones mayormente aceptadas. Según los Estados Unidos una enfermedad que afecte menos de 200.000 individuos es considerada una enfermedad rara[3]. Mientras que en Europa ésta debe afectar uno de cada 2.000 individuos[4] para pertenecer a dicha categoría.

Para los propósitos de esta tesina vamos a tomar la definición propuesta por Orphanet[6] y respaldada por el Parlamento Europeo según la cual una enfermedad rara es aquella que afecta uno de cada 2.000 individuos.

Relevancia de las enfermedades raras.

Si bien las enfermedades raras se caracterizan por afectar un número reducido de individuos, en conjunto estas patologías suponen un impedimento a la salud pública. La *Organización Mundial de la Salud* estima alrededor de 400 millones de casos de enfermedades raras[2] a nivel mundial, superando los casos de cáncer y de SIDA.

Cualquier terapia clínica requiere determinar la naturaleza de la patología para adecuar el tratamiento a las condiciones del paciente. Sin embargo, la amplia mayoría de pacientes con enfermedades raras y sus familias deben pasar por lo que se conoce como la "odisea del diagnóstico".

Como consecuencia de la falta de diagnóstico, varios pacientes sufren consecuencias indeseadas producto del mal uso de las drogas, de la postergación de algún tratamiento o de la toma de acciones incorrectas para mejorar la calidad de vida del paciente. Pero no solo el individuo se ve afectado, estudios han mostrado que el principal motivo de sufrimiento para sus familiares proviene de la incertidumbre inherente al proceso de diagnóstico[11].

Un caso conocido es el de Mila Makovec (8 años)[56] quien sufre una condición neurodegenerativa producida por una mutación recesiva en el gen MFSD8 conocida como la enfermedad de Batten. Esto hizo que la madre de Mila deba contactar decenas de médicos y recolectar 3 millones de dólares en donaciones para pagar por la investigación que desarrollara una cura para su hija[57]. Otro es el caso de Sergio Isla, un joven español con síndrome de Dravet que mostró síntomas anormales a los tres meses de edad. El paciente tuvo que esperar 5 meses para ser diagnosticado, periodo en el cual le prescribieron un medicamento contraindicado para su enfermedad. Este le produjo docenas de crisis epilépticas al día con un daño cerebral inconmensurable[7]. De haber podido acceder a un diagnostico temprano, Sergio y su familia no hubiesen pasado por una experiencia tan desagradable.

La "odisea del diagnostico" es el problema por excelencia cuando hablamos de enfermedades raras. Un paciente promedio debe esperar 6 años para ser diagnosticado y visita un estimado de 7 especialistas durante el proceso[8]. Ciertos estudios han demostrado que con el aumento de las visitas al medico disminuye la posibilidad de obtener un diagnostico claro[9], lo que resulta en un 50% de pacientes sin diagnosticar[39]. Las características biológicas que subyacen a las enfermedades raras hacen muy difícil su diagnóstico.

Para tratar este problema se ha impulsado la creación de una serie de consorcios dedicados a facilitar el diagnostico temprano de estas enfermedades. De entre ellos se destaca la formación del *Consorcio Internacional para la Investigación en Enfermedades Raras* (IRDiRC)[12], el cual tiene como objetivo proporcionar un diagnóstico confiable a cualquier paciente con una patología rara en un periodo máximo de un año[13]. Estas organizaciones, en conjunto con fuertes entes inversores[14] y el desarrollo de las nuevas tecnologías están permitiendo desentrañar los fundamentos biológicos de estas enfermedades que todavía no se comprenden en su totalidad.

Características de las enfermedades raras.

La amplia mayoría (80%) de las enfermedades raras conocidas tienen base genética[15], pero no todas las patologías de base genética son enfermedades raras. La literatura moderna tiende a definir las enfermedades genéticas en un gradiente de complejidad según el peso y número de variantes que participen en el desarrollo de la misma[16]. Así pasamos de enfermedades complejas (también conocidas como comunes) en un extremo, que se presume son causadas por un conjunto de variantes independientes con baja carga patogénica. Y las enfermedades monogénicas (también conocidas como mendelianas o raras) en el otro extremo, que se presume son causadas por una o pocas variantes que contribuyen en gran medida al fenotipo observado.

Hoy en día se conocen entre 5.000 y 8.000 diferentes tipos de enfermedades raras[2], muchas de ellas de carácter crónico. Las mismas suelen además desarrollar síntomas en edades muy tempranas, 50% de ellas afectando niños y con una tasa de sobrevida muy baja.

Gran parte de la variación fenotípica observada puede atribuirse a una o pocas variantes (SNPs). Esta característica hace que las enfermedades raras sean muy difíciles de estudiar sin análisis genéticos. Por ello el diagnóstico genético utilizando tecnologías de secuenciación masiva ha adquirido gran relevancia recientemente.

Surgimiento de NGS

La secuenciación masiva accesible como la conocemos surge por una combinación de avances tecnológicos que se fueron dando en un período de 40 años. Desde los inventos Maxam & Gilbert[17] y Sanger & colegas[18] en 1977, la demanda creció exponencialmente llevando a la instalación de galpones enteros con secuenciadores automáticos[19] y la publicación del primer genoma humano en 2001[20], con un costo de 3.000.000.000 USD. Éste se secuenció en parte con el primer secuenciador que utilizaba tecnología shotgun, Celera.

A diferencia de los secuenciadores de primera generación (ej. Sanger) que secuencian siguiendo el orden de la hebra, la tecnología shotgun busca secuenciar la mayor cantidad de fragmentos posibles para luego ensamblar la hebra de ADN. Nuevas tecnologías empezaron a surgir con el objetivo de secuenciar la mayor cantidad de fragmentos al menor costo posible. La tecnología 454 de Roche fue el primer secuenciador en usar la señal emitida durante la síntesis de ADN. Sin embargo, poco después Illumina desarrolló una tecnología basada en síntesis que permite secuenciar mayor cantidad a menor precio. Hasta el día de hoy Illumina continúa siendo la tecnología mas usada por su relación entre calidad del base-calling, profundidad de secuenciado y precio. El punto débil de las tecnologías basadas en síntesis es el largo de las secuencias que pueden producir. La calidad de la polimerasa disminuye a medida que va sintetizando la hebra, por lo que se pierde mucha precisión al acercarse a las 100 bases secuenciadas. Para complementar este problema han surgido las tecnologías de tercera generación como Nanopore y PacBio que buscan generar secuencias largas a cuestas de una menor precisión.

Hoy en día podemos secuenciar un genoma humano entero por solo 1.000 dólares[21] (de reactivos) e incluso hacernos con datos públicos gracias a proyectos internacionales[22-23]. La creciente cantidad de información producida por los secuenciadores y la reducción de costos han hecho de la genómica el área computacionalmente mas demandante del mundo. Estimaciones de 2015 la posicionaban por encima de fuentes masivas de datos como la astronomía, YouTube y Twitter[24]. Esto ha impulsado el desarrollo de herramientas informáticas eficaces en procesar datos biológicos.

Aplicaciones bioinformáticas a datos de secuenciación masiva

Para pasar la información nucleotídica a archivos digitales, los secuenciadores mas usados (ej. Illumina) codifican señales biológicas (en general) en un formato de texto plano llamado “fastq”. Este está formado por

un conjunto de secuencias cortas (lecturas) con sus respectivos identificadores y medidas de calidad, de éste derivan todos los análisis bioinformáticos aplicados a datos genómicos. Dicho archivo se procesa de manera estandarizada para obtener las variantes en el genoma. El primer paso consiste en remover las lecturas de baja calidad. El segundo paso es mapear las lecturas que hayan pasado el filtro de calidad. Para esto existen genomas de referencia como hg19, que son representaciones haplotípicas y genéricas que buscan asemejarse al genoma humano. Las lecturas encuentran su posición en dicho genoma a partir de un algoritmo de mapeado (ej. bwa[73]). Esto conduce a un archivo sam (o bam) a partir del cual se hace el denominado "variant-calling". En esta tercera etapa se ven los cambios a nivel nucleotídico entre el genoma secuenciado y la referencia. Así se obtiene archivo VCF (Variant Calling Format) que especifica el cromosoma, posición y cambio nucleotídico observado para cada variante y cada individuo procesado. El cuarto y último paso consiste en anotar dichas variantes con diversas características biológicas. Para esto se han desarrollado bases de datos de frecuencia poblacional (gnomAD[69], 1000G[62]), modo de herencia (OMIM[70]), cambio aminoacídico (PolyPhen2[50]), conservación evolutiva (GERP[68]), caracterización de genes (GenBank[58]), dominios proteicos (Pfam[59]), variación genética (HapMap[60]), entre otras. También podemos encontrar bases de datos de proyectos públicos (UKBioBank[61]) y organizaciones privadas (HGMD[63], 23andMe[64]). Existen tres instituciones gubernamentales que agrupan la amplia mayoría de las bases de datos biológicas disponibles. Estas son el Centro Nacional de Información Biotecnológica (NCBI[65]) de los Estados Unidos, el Instituto Europeo de Bioinformática (EBI[66]) de la Unión Europea y el Instituto Nacional de Genética (NIG[67]) de Japón.

La base de datos ClinVar busca definir la carga patogénica (Significancia Clínica) para la mayoría de las variantes conocidas. Para ello divide su anotación en 13 categorías, estas son: Benigna, Posiblemente benigna, Significado incierto, Posiblemente patogénica, Patogénica, Respuesta a droga, Asociación, Factor de riesgo, Protege, Afecta, Información conflictiva, Otro y No proporcionado. Las 5 primeras son las mas utilizadas siguiendo las pautas establecidas por el Colegio Americano de Genética Medica y Genómica (ACMG) para la interpretación clínica de variantes[71]. Se trata de un recurso masivo al que cualquiera puede subir información. Esto genera anotaciones inconsistentes por lo que requiere un amplio proceso de curado y filtrado manual de las variantes antes de poder usarla. Gracias a los requerimientos para subir una variante, cada anotación de patogenicidad esta respaldada por una escala de confianza del 0 al 4 (Criterio de afirmación). Se le asigna 0 a las variantes sin interpretación, 1 a aquellas con una sola interpretación o múltiples interpretaciones conflictivas, 2 a las variantes con dos o mas interpretaciones idénticas y 3 a las variantes revisadas por un panel de expertos. Esto permite filtrar la información para conservar únicamente entradas confiables.

Una aplicación directa para los datos de ClinVar es el estudio de variantes asociadas a patologías con base genética. Para ello se deben anotar las variantes con las bases de datos mencionadas anteriormente. Un anotador muy utilizado y gratuito para investigación es ANNOVAR, este permite anotar variantes de manera muy sencilla con las bases de datos que el usuario especifique.

Como se mencionó anteriormente, las enfermedades genéticas son difíciles de diagnosticar para la medicina clásica, por lo que el análisis bioinformático de variantes en el genoma/exoma de pacientes no diagnosticados ha tomado gran relevancia. Estas aproximaciones permitieron identificar en 2015 los genes responsables del 50% de las enfermedades raras conocidas[25]. Sobre esta línea en los últimos años se han llegado a secuenciar centenares de miles de genomas/exomas con el objetivo de identificar nuevas variantes patogénicas. A pesar de este esfuerzo, hoy en día estamos lejos de identificar las variantes causales para la mayoría de las enfermedades conocidas. Dado que las regiones no codificantes son mas abundantes, mas divergentes y se han secuenciado en menor cantidad, es muy probable que el bajo porcentaje de enfermedades diagnosticadas se deba a lo poco que se han estudiado las variantes no-codificantes del genoma humano.

Regiones no-codificantes:

Como se mencionó anteriormente, muchas enfermedades mendelianas no pueden ser diagnosticadas usando datos de NGS en regiones codificantes. Surge así la necesidad de estudiar el efecto fenotípico producido por cambios en regiones no-codificantes del genoma humano.

Estas regiones representan aproximadamente el 98.5% de la información genética. Si consideramos que en promedio puede haber 3 millones de variantes en un genoma humano con respecto a la referencia, aquellas en regiones no-codificantes representan al menos 2.955.000 mutaciones. Aún siendo 66 veces más abundantes, el entendimiento general de las regiones no traducidas del genoma es muy limitado. Recién se está comenzando a entender el funcionamiento de los ARNs no-codificantes y los pseudogenes, a quienes se los asocia cada vez mas con procesos biológicos que regulan la expresión génica[40-42]. A su vez, existen elementos reguladores en cis- (intrones, UTRs, elementos proximales y promotores) y elementos reguladores en trans- (elementos distales) que controlan la transcripción de uno o varios genes. Por último, encontramos regiones repetidas o ADN "basura" cuya función se desconoce, lo cual no implica falta de función biológica. El proyecto ENCODE ha demostrado que aproximadamente el 80% de nuestro genoma cumple una función bioquímica[55], por lo que este ADN "basura" bien podría ser importante.

En resumen, un cambio en regiones no-codificantes puede ser suficiente para alterar la expresión de un gen afectando el proceso biológico en el que su producto actúa. Varios estudios están remarcando el rol de estas variantes en el desarrollo de enfermedades monogénicas y complejas[43-44], particularmente las de baja frecuencia. Considerando la enorme cantidad de variantes no-codificantes (NCV) que podemos encontrar en un genoma humano y los efectos que pueden causar, definitivamente tienen que ser consideradas a la hora de realizar diagnósticos de ER basados en NGS.

Afortunadamente, la baja de precios está dando mayor lugar al análisis de genoma completo. Esto pone al alcance estudios poblacionales de variantes lo cual aumenta el poder estadístico de nuestros análisis, permite identificar variantes raras o ultra-raras (singletons) y aproximar la priorización de las mismas tomando en cuenta la ancestría particular de cada individuo. Sin embargo, el estudio de variantes no-codificantes tiene un grado extra de complejidad. Como se mencionó anteriormente, las bases de datos con las cuales se anotan las variantes se basan mayoritariamente en cambios aminoacídicos. Por lo tanto, hay mucha información disponible para caracterizar variantes en regiones codificantes y poder clasificarlas. Este no es el caso en SNPs de regiones no-codificantes ya que no cuentan con información correspondiente a cambios aminoacídicos lo cual hace difícil cuantificar su relevancia funcional. Los predictores disponibles para utilizar en dichas regiones son su conservación evolutiva, frecuencia poblacional y ciertos scores integrales que usan varias bases de datos para dar un estimado de su patogenicidad. Al ser mas difíciles de clasificar, la amplia mayoría de variantes no codificantes del genoma humano están anotadas como significado incierto. El bajo numero de predictores que se pueden utilizar y su abundancia en el genoma lo cual genera set de datos masivo representan un problema para los métodos clásicos de análisis bioinformático como el filtrado de variantes. Sin embargo, el gran numero de variantes no-codificantes existentes es una ventaja al intentar predecir la significancia clínica de las variantes utilizando abordajes basados en Aprendizaje Automático (AA), los cuales se benefician con una mayor abundancia de datos.

Además, la anotación en regiones no-codificantes suele estar fragmentada por lo que se encuentra un gran número de valores ausentes (NAs) en cualquier base de datos que se quiera analizar. Una solución muy utilizada para hacer frente a este problema es reemplazarlos por el promedio de la variable (columna) en la que se encuentran. Sin embargo, esta aproximación muchas veces no tiene sentido biológico, por lo que resulta de suma importancia encontrar métodos alternativos para la imputación de NAs en datos biológicos que derivan de la anotación de cada variante.

Los valores ausentes se clasifican en cuatro tipos según su distribución y estructura. Por un lado, tenemos los datos estructuralmente ausentes (SMD). Éstos no deberían estar presentes y dependen de la naturaleza de nuestros datos[72]. En estos casos el mejor abordaje es simplemente eliminar la columna que contiene dichos datos. Por otro lado, tenemos los datos ausenten de manera completamente aleatoria (MCAR). La estructura

de éstos NAs es independiente de los demás datos de la tabla, sean observados o no observados. Este tipo de valores afecta el poder estadístico de la muestra ya que no generan ningún tipo de patrón que pueda resultar informativo para el clasificador[72]. También existen los datos ausentes aleatoriamente (MAR), en los cuales la ausencia de un valor depende en cierta medida de los valores observados[72]. Y por último tenemos los valores ausentes no aleatorios (MNAR), los cuales dependen de factores que no son tomados en cuenta para el relevamiento de los datos.

Cualquier estudio que involucre anotaciones en regiones no codificantes se va a ver enfrentado al problema de los valores ausentes. Por ello resulta importante encontrar algoritmos que interpreten bien cada set de datos particular.

Machine Learning

El Aprendizaje Automático o Machine Learning (ML) es el estudio de algoritmos y modelos estadísticos que permiten realizar una tarea basándose en patrones e inferencia sobre los datos[26]. Los modelos generados nos permiten obtener nuevas perspectivas sobre situaciones reales o hacer predicciones sobre sus posibles desenlaces.

El término Aprendizaje Automático agrupa un conjunto de algoritmos capaces de identificar patrones comunes en grandes cantidades de datos. Éstos suelen agruparse en métodos supervisados, métodos no-supervisados y métodos de aprendizaje por refuerzo.

La principal característica del entrenamiento por Aprendizaje Supervisado es que busca modelar los patrones que encuentra en los datos según el resultado que estos tengan asignados. Se los suele distinguir entre regresiones, cuyas predicciones son cuantitativas o continuas y clasificaciones, cuyas predicciones son categóricas o cualitativas. Algunos algoritmos conocidos son los modelos basados en árboles (Arboles de Decisión, Bosques Aleatorios), los procesos Gaussianos, las Maquinas de Soporte de Vectores, los modelos de vecinos cercanos (KNN) y las redes neuronales (Perceptrones Multicapa, RN Profundas, RN Convolucionales).

Los Arboles de Decisión se conforman por varios nodos de probabilidad en los que se asigna la probabilidad de ocurrencia de dos eventos que derivan de un nodo de decisión y se dirigen a otro nodo de probabilidad por ramificaciones alternativas. Estos nodos de probabilidad buscan separar los datos ingresados dando los resultados en el nodo terminal. Los bosques aleatorios son una alternativa a este proceso en el que cada árbol se entrena usando un subset de los predictores disponibles. Sus predicciones en el nodo terminal se promedian para dar un resultado único.

Las maquinas de Soporte de Vectores simples buscan agrupar los datos separándolos por un vector. Resultan muy útiles cuando se le aplican diferentes kernels al vector que permiten separar los datos en un plano multidimensional.

Un área que dentro del Aprendizaje Automático que ha generado mucho interés es el Aprendizaje Profundo o utilizando Redes Neuronales. Éstas simulan la estructura de neuronas del cerebro humano al requerir un umbral de activación para enviar la señal a la siguiente capa de neuronas. La generación de diversas capaz neuronales consecutivas es lo que le debe su nombre de Aprendizaje Profundo.

Por otro lado, el entrenamiento por Aprendizaje No-Supervisado no cuenta con una variable resultado que determine la naturaleza de los datos. Por lo que el modelo se genera buscando patrones que permita agrupar la información basados únicamente en las características intrínsecas de los mismos. Algunos algoritmos conocidos son los mapas topográficos, los de cuantificación vectorial y los de maximización de las expectativas.

Por último, el Aprendizaje Reforzado utiliza la interacción de un agente con el medio para definir el modelo. El agente es recompensado por una acción que se define según la situación que se quiera modelar. En este caso el algoritmo más utilizado es el Q-learning.

El crecimiento de las capacidades de cómputo ha permitido que más áreas apliquen Inteligencia Artificial a sus proyectos, y dentro de ella hagan uso del Aprendizaje Automático. Las más ampliamente desarrolladas son la prevención de fraude, la visión computacional, el procesamiento de lenguaje natural y recientemente la biología computacional. Estas técnicas requieren grandes cantidades de datos, por lo que el surgimiento del NGS ha permitido poner en practica dichas aproximaciones en el área de la genómica. En este proyecto nos vamos a centrar en las aplicaciones del Aprendizaje Supervisado.

Aplicaciones de AA a grandes bases de datos biológicos

Entre las principales aplicaciones de AA a datos biológicos están el diagnóstico con imágenes médicas y el análisis de datos de secuenciación. Cualquiera de estas áreas cuenta con datos masivos que permiten entrenar los algoritmos. En el caso de los datos de secuenciación genómica (DNA-seq), éstos suelen usarse para identificar regiones como promotores[27], potenciadores[28], sitios de splicing[29] y demás[30]. También existen aplicaciones que involucran otros tipos de datos, como transcriptomas (RNA-seq) o microarrays. Estas toman información de la expresión génica para identificar biomarcadores que permitan diferenciar patologías[46-47]. Otras utilizan los datos de secuenciación de la cromatina (ChIP-seq) para identificar nuevos elementos funcionales a través de algoritmos no-supervisados[45]. Y otras agrupan diferentes tipos de información biológica para anotar genes[31-32] e inferir patrones de expresión génica[33-34]. Por último, existen aplicaciones que integran datos de secuenciación, imágenes médicas e historiales clínicos para buscar un diagnóstico a pacientes con enfermedades raras. Un ejemplo es la plataforma Dx29[53] impulsada por Julián Isla, familiar de un paciente con enfermedad rara. Esta herramienta utiliza visión computacional, procesamiento de lenguaje natural y priorización de variantes en exomas para proporcionar un diagnóstico para individuos con enfermedades raras. De momento, en su fase "beta" la herramienta ha sido capaz de diagnosticar pacientes con una eficacia del 80%[54].

Recientemente, se comenzó a utilizar la información disponible en las bases de datos para desarrollar herramientas de AA destinadas a caracterizar las variantes del ADN. Esto es posible gracias a los consorcios públicos que emprenden proyectos de secuenciación a nivel poblacional como son el 1000G, Genomics England, 100.000 Genomes Project (Australia), Precision Medicine Initiative (Estados Unidos), GenomeAsia 100K. Dichos consorcios suben a bases de datos públicas la frecuencia poblacional de cada variante. Esta información es muy valiosa ya que genera una referencia a partir de la cual comparar e identificar variantes nuevas que puedan estar relacionadas con enfermedades. Dicha información combinada con predictores de patogenicidad y conservación evolutiva impulsan las aplicaciones de Aprendizaje Automático.

Las herramientas desarrolladas hacen uso integral de esta información para encontrar patrones característicos que permitan diferenciar variantes. Una vez entrenados, los modelos pueden ser usados para predecir posibles variantes patogénicas en pacientes no diagnosticados.

Cada herramienta emplea su propia aproximación para priorizar variantes y todas consiguen muy buenas medidas de precisión. Algunas entrenan algoritmos basados en árboles usando sets de entrenamiento altamente desbalanceados con variantes patogénicas como datos positivos y posiciones divergentes del genoma humano como datos negativos (REVEL[35], NCBoost[36] y ReMM[37]). Otras usan únicamente la conservación evolutiva de los nucleótidos para determinar qué tan deletéreas o neutrales pueden ser las variantes en dicha posición (CADD[38]), en este entrenan un modelo de regresión logística a partir de un set masivo de 15 millones de SNPs y 1.8 millones de Indels. Otras herramientas entrenan una red neuronal profunda usando en parte datos simulados (DANN[48]). Por último, está el ejemplo de RegBase[49], una herramienta que utiliza una aproximación similar a la presentada en este trabajo ya que toma variantes anotadas como patogénicas para su set de

entrenamiento positivo y variantes anotadas como benignas para sus datos negativos, entrenando así un algoritmo basado en árboles con aumento de gradiente (Gradient Boosting Trees).

De todas las herramientas que hacen uso del Aprendizaje Automático para priorizar variantes, la amplia mayoría se dedica a identificar mutaciones patogénicas en regiones no-codificantes del genoma. Como ya mencionamos, estas variantes son muy abundantes y difíciles de caracterizar. En ellas no se pueden usar los predictores altamente informativos basados en cambios peptídicos como PolyPhen2[50], SIFT[51], PROVEAN[52] y demás. Por ello, las herramientas que integran numerosas propiedades biológicas de las NCV en una sola métrica son una buena manera de caracterizar dichas variantes.

Proyecto URUGENOMES

En 2014, el Institut Pasteur de Montevideo puso en marcha el proyecto Genoma de los Uruguayos (URUGENOMES). Este consta de tres fases, la primera con un enfoque antropológico, busca ilustrar el genoma del pueblo Charrúa. La fase intermedia está destinada a una caracterización general de la población uruguaya. Y la tercera fase, en la que se enmarca este proyecto de grado, busca diagnosticar pacientes uruguayos que sufren de algún tipo de patología no diagnosticada. Para ello se secuenció el genoma de 30 pacientes y familiares sufriendo algún tipo de enfermedad rara.

Para alcanzar el objetivo propuesto en la fase tres, es necesario poner a punto un pipeline interpretable por médicos genetistas. El estudio de variantes en regiones codificantes se lleva a cabo con una aproximación estándar pero efectiva. Esta consiste en priorizar variantes por métodos bioinformáticos clásicos y luego analizarlas en detalle con un genetista clínico. Este trabajo de grado busca complementar dicho pipeline desarrollando una aproximación que permita priorizar variantes en regiones no-codificantes para entregar un set reducido que pueda ser analizado en detalle con un genetista clínico.

Una vez puesto a punto, el pipeline desarrollado se aplicará a cada paciente secuenciado en la fase tres del proyecto URUGENOMES con el objetivo de identificar variantes causales tanto en regiones codificantes como en regiones no-codificantes del genoma.

Objetivo general

Desarrollar y evaluar un clasificador basado en métodos de Aprendizaje Automático para priorizar variantes no-codificantes del genoma humano.

Objetivos específicos

1. Definir juego de datos de entrenamiento basado en la base de datos ClinVar.
2. Imputar datos faltantes en las anotaciones de cada variante.
3. Evaluar el desempeño de tres particiones sobre el set de entrenamiento.
4. Aplicar diferentes algoritmos de Aprendizaje Automático con complejidad creciente al juego de datos y evaluar la precisión de cada uno.
5. Aplicar el algoritmo con mayor desempeño a los datos del proyecto URUGENOMES.

Metodología

Esquema del trabajo

El esquema de trabajo se alinea con los objetivos específicos detallados en la introducción, dividiéndose en cinco etapas: i) obtención de datos, ii) imputación, iii) entrenamiento, iv) evaluación v) predicción de pacientes de URUGENOMES.

Sobre esta línea, el primer paso es obtener información de la base de datos ClinVar. Esta nos proporciona una anotación referente al efecto fenotípico (Significancia Clínica) observado en cada variante. Una vez curados los datos, se va anotar cada variante usando el programa de acceso libre para investigación ANNOVAR. Allí se obtiene información sobre la región genómica en la que se encuentra cada variante, sea codificante (exones) o no-codificante (intrones, sitios de splicing, UTRs, regiones intergénicas, etc). Además, consulta bases de datos terciarias para anotar cada variante, por lo que también nos proporciona información sobre la frecuencia (1000G, Kaviar, AF), conservación evolutiva (PhyloP, PhasCons, SiPhy) y grado de patogenicidad predicha (CADD, Eigen, GWAVA).

El segundo paso consiste en imputar valores ausentes en la anotación de cada variante. Esto es particularmente importante cuando se trabaja con variantes no-codificantes ya que la anotación para éstas es muy pobre en las bases de datos. Por ello, realizamos un estudio comparativo con el objetivo de identificar un algoritmo óptimo para inferir datos genómicos en dichas regiones. Previo a la imputación se filtran las variantes con menos del 30% de sus valores anotados y luego se aplica un algoritmo basado en vecinos más cercanos (KNN) para inferir los valores ausentes. Las variables (columnas) cuya imputación no haya sido precisa van a ser completadas entrenando un modelo de red neuronal (perceptrón multicapa simple) que permita predecir mejor sus valores ausentes.

El tercer paso consiste en entrenar algoritmos de aprendizaje supervisado utilizando tres escenarios o sets de entrenamiento diferentes. En este paso se busca definir el escenario óptimo con el cual entrenar los modelos y los hiperparámetros a utilizar en cada algoritmo. Las variantes no utilizadas en el entrenamiento se reservan para evaluar el desempeño de cada modelo en la siguiente etapa.

En el cuarto paso se van a predecir los modelos entrenados utilizando las variantes reservadas para evaluación. El desempeño de los algoritmos se va a analizar con métricas comparativas como Sensibilidad y Especificidad.

Por último, el modelo seleccionado con hiperparámetros optimizados va a ser utilizado para priorizar variantes en el genoma de pacientes uruguayos, dentro del marco de la fase tres del proyecto URUGENOMES. Para poder ser utilizados por el clasificador, dichos archivos van a tener que ser anotados e imputados con el mismo protocolo mencionado anteriormente.

1. Obtención de datos

Las variantes en el genoma humano fueron obtenidas de la base de datos ClinVar, del archivo (.txt) subido el 6 de Marzo de 2019.

Ésta forma parte de las bases de datos creadas por el Centro Nacional de Información Biotecnológica (NCBI) de los Estados Unidos. Al ser un repositorio público cualquier investigador puede subir información allí, esto llevó a que para Marzo de 2019 contara con 976.907 variantes, 490.571 de las cuales corresponden al genoma de referencia hg19 (GRCh37).

El proceso de subir una variante a ClinVar implica varios puntos de control que aseguran la veracidad de las anotaciones allí depositadas. En primer lugar, el investigador debe proporcionar información general de la variante según las normas de la Sociedad de Variación en el Genoma Humano (HGVS). Éstas son su posición genómica, largo, número de copias, cambios nucleotídicos involucrados, detalles del gen/proteína modificado/a o su distancia al gen más cercano, las consecuencias funcionales del cambio, nombre de la variante e identificación en otras bases de datos públicas. En segundo lugar, se debe especificar la patología o fenotipo que la mutación produce y una interpretación para la misma. Por último, el investigador debe presentar evidencia apoyando dicha anotación. Esta evidencia puede derivar de trabajos *in-silico*, pero debe venir

acompañada por experimentos de mesada u observaciones en pacientes. Ésta incluye el número de observaciones que asocian la variante con dicha significancia clínica, su modo de herencia y presencia de segregación o historia familiar.

ClinVar clasifica la significancia clínica de las variantes en 15 grupos, estos son *Affects*, *association*, *Benign*, *conflicting data from submitters*, *Conflicting interpretations of pathogenicity*, *drug response*, *Likely benign*, *Likely pathogenic*, *no interpretation for the single variant*, *not provided*, *other*, *Pathogenic*, *protective*, *risk factor*, *Uncertain significance* y combinaciones de las mismas. Cada variante puede estar anotada por más de un investigador, por lo que las combinaciones de etiquetas y el numero de veces que cada etiqueta se asignada a una variante, dan una idea de su patogenicidad.

Para focalizar el objetivo de nuestro clasificador redujimos el número de etiquetas a *Pathogenic*, *Likely pathogenic*, *Uncertain significance*, *Likely benign* y *Benign*. Para esto, las variantes que están anotadas con combinaciones de etiquetas fueron reducidas a una de estas las 5 categorías. Dichas combinaciones se asignaban a la etiqueta que tenga mayor representación. Las variantes anotadas como Uncertain significance no tuvieron que ser reducidas a partir de combinaciones ya que en la versión obtenida no presentan anotación ambigua.

Dado que las variantes en regiones no-codificantes son mayormente de significado incierto, es importante poder evaluar el desempeño de nuestro clasificador en este tipo de mutaciones. Para ello descargamos el archivo extendido (.xml) de ClinVar, subido el 3 de Abril de 2019. En éste obtuvimos 536 variantes en regiones no-codificantes, previamente anotadas cómo *Uncertain significance* a las que luego se le asignó la etiqueta *Pathogenic* o *Benign*. De éstas, 43 mostraron estar relacionadas con algún tipo de patología (*Pathogenic*) y 493 mostraron evidencia de lo contrario (*Benign*). Al haber sido en primer lugar definidas como *Uncertain significance* suponemos que estas variantes presentan características similares a las variantes con significado incierto en ClinVar. La segunda etiqueta nos indica que biológicamente pueden presentar indicios de patogenicidad. Esto nos permite definir la predicción correcta a la que debe llegar el clasificador y así evaluar su desempeño sobre dicho tipo de variantes.

Luego de filtrar la base de datos y agregar las variantes anotadas como Uncertain significance - Pathogenic/Benign contamos con 310.227 SNPs en regiones codificantes y 97.736 SNPs en regiones no-codificantes (Tabla 1).

	Codificantes	No-Codificantes
Benign	15.768	13.709
Likely benign	49.448	30.670
Pathogenic	47.717	6.048
Likely pathogenic	19.531	5.023
Uncertain significance	149.785	41.750
Uncertain significance / Benign	24.047	493
Uncertain significance / Pathogenic	3.931	43
Total	310.227	97.736

Tabla 1. Distribución de las anotaciones de significancia clínica en ClinVar luego del curado de datos.

Anotación:

Una vez obtenidas las variantes con sus respectivas significancias clínicas procedemos a anotarlas con scores de patogenicidad, frecuencias y medidas de conservación evolutiva. Para este paso se utilizó la herramienta ANNOVAR[74], de acceso libre para investigación.

ANNOVAR utiliza un archivo de texto plano como entrada en el que cada variante tiene definido su cromosoma, sitio de inicio, sitio de terminación, nucleótido de referencia y nucleótido observado, en este orden. La herramienta cuenta con dos funciones principales, una de ellas (*annotate_variation*) que puede realizar tres tipos de procesos (*gene-based*, *region-based* y *filter-based*) según el archivo de salida particular que necesite el usuario. La otra, (*table_annovar*) permite integrar los tres procesos, ésta devuelve un archivo de texto plano delimitado por tabulaciones en el que cada columna corresponde a una base de datos especificada por el usuario. *table_annovar* es la función más utilizada en ANNOVAR ya que consigue una anotación robusta y fácil de interpretar.

Para poder correr la herramienta primero se deben descargar al disco local los archivos Generic Feature Format (GFF) de cada una de las bases de datos que se quiera utilizar para anotar los SNPs, según la referencia que se esté usando. Consultando estos archivos, ANNOVAR es capaz de identificar variantes intrónicas, exónicas, intergénicas, 5'/3' UTRs, en sitios de splicing y en sitios 1 kb upstream o 1 kb downstream de la región traducida. Para variantes exónicas reporta el gen en el que se encuentra, los cambios aminoacídicos que produce y formación o eliminación de un codon de terminación/iniciación. Para variantes no exónicas reporta el transcripto más cercano al sitio de la variante. Y por último ANNOVAR incluye la anotación de diferentes bases de datos especificadas por el usuario. Éstas pueden ser medidas de patogenicidad basadas en cambios aminoacídicos, frecuencias poblacionales, medidas de conservación evolutiva y demás.

Se utilizó la función *table_annovar* de ANNOVAR para anotar las variantes obtenidas de ClinVar. Para ello se generó un archivo de texto plano con dichas variantes indicando cromosoma, sitio de inicio, sitio de terminación, nucleótido de referencia y nucleótido observado. Los repositorios interrogadas fueron refGene (v. 16-4-2018) con el proceso de *gene-based annotation*, cytoBand (v. 20-11-2018) con el proceso de *region-based annotation* y los repositorios ExAC (v. 20-11-2018), avsnp (v. 20-11-2018), dbnsfp (v. 20-11-2018), regsnpintron (v. 8-5-2019), GWAVA (v. 8-5-2019), gnomAD (v. 8-5-2019), Intervar (v. 8-5-2019), Kaviar (v. 8-5-2019), ClinVar (v. 8-5-2019) y Eigen (v. 20-8-2019) con el proceso de *filter-based annotation*. Algunos de éstos agrupan información de varias bases de datos, por lo que durante el proceso de anotación se interrogó SIFT, PolyPhen2-HVAR/HDIV, Meta-SVM/LR, LRT, FATHMM, PROVEAN, VEST3, MutationTaster, MutationAssessor, FATHMM-MKL, fitCons, ExAC, gnomAD, Kaviar, GERP, phyloP, phastCons, SiPhy, regSNP, InterVar, CADD, DANN, GWAVA y Eigen. Cada una de éstas utiliza diferentes propiedades biológicas para caracterizar la mutación, por lo que al agruparlas obtenemos perfiles que el algoritmo de Aprendizaje Automático puede utilizar para desvelar patrones característicos en variantes patogénicas. Procedemos a introducir brevemente cada base de datos.

Sorting **I**ntolerant **F**rom **T**olerant (SIFT) utiliza las propiedades fisicoquímicas de los aminoácidos para evaluar si el cambio de un residuo puede afectar la función de una proteína[75]. Para ello generan alineamientos múltiples a partir de los datos obtenidos por Ng, P. C. & Henikoff, S.[80], asumiendo que las regiones mas conservadas van a ser menos tolerantes a cambios y por lo tanto un cambio aminoacídico allí va a tender a afectar el correcto funcionamiento de la proteína.

Polymorphsim **Phen**otyping v**2** (PolyPhen2) permite anotar SNPs no sinónimos en regiones codificantes, por lo que evalúa los efectos de un cambio aminoacídico en una proteína a través de la homología de secuencias. Al igual que la herramienta anterior, PolyPhen2 crea alineamientos múltiples para ver las regiones más conservadas en una cadena polypeptídica. Para ello usan las bases de datos HumDiv y HumVar[81] generadas a partir de UniProtKB. Esta herramienta además integra características estructurales de la proteína en cuestión cómo superficie accesible, hidrofobicidad y demás[76].

Likelihood **R**atio **T**est (LRT) usa alineamientos de 32 especies de vertebrados. Aplicando el test del cual deriva su nombre es capaz de identificar mutaciones que irrumpan aminoácidos conservados[78].

Functional **A**nalysis **T**hrough **H**idden **M**arkov **M**odels (FATHMM[79]) utiliza modelos HMM generados a partir de variantes anotadas como dañinas (DM) en la Base de Datos de Mutaciones en Genes Humanos (HGMD)[82] y sitios presuntamente neutros en UniProt[83]. Dichos modelos HMMs se usan para interrogar UniRef90, SUPERFAMILY, Pfam, SwissProt y TrEMBL para calcular el posible efecto de un cambio en cada posición de la cadena de Markov escondida.

Protein **V**ariation **E**ffect **An**alyser (PROVEAN) genera agrupamientos de secuencias con más de 75% de identidad según BLAST. Mantiene los 30 agrupamientos más similares a secuencias previamente alineadas y los usa como referencia para calcular medidas de patogenicidad[84], según el efecto funcional que puedan tener las regiones mas conservadas.

Mutation Taster busca caracterizar mutaciones sinónimas, cambios de un solo amino ácido y alteraciones que afecten la secuencia completa de una proteína. Se generó relevando información de varias bases de datos y de la literatura para medir la conservación evolutiva de los sitios, perdida en la función de la proteína y cambios en sitios de splicing o vías de señalización que puedan afectar la abundancia de ARNm[86]. Con estos datos infiere el afecto de una mutación sobre las propiedades funcionales de una proteína.

Mutation Assessor genera agrupamientos específicos de subfamilias de proteínas para determinar qué cambios son conservados y qué cambios son específicos de una subfamilia de proteínas[87]. A partir de éstos infiere con qué confianza una proteína forma parte de una subfamilia. A partir de este grado de confianza y las propiedades de la subfamilia genera estimaciones numéricas de los efectos que pueda tener cada cambio sobre la estabilidad de la proteína y su interacción con otras moléculas.

Meta-SVM/LR surge a partir de un estudio que buscaba determinar el mejor método para predecir patogenicidad en regiones codificantes del genoma humano[77]. En el proceso desarrollaron un modelo de Maquina de Soporte de Vectores (SVM) y otro de Regresión Linear (LR) utilizando nueve diferentes predictores (SIFT, PolyPhen2-HVAR, PolyPhen2-HDIV, LRT, Mutation Assessor, FATHMM, GERP++, PhyloP, SiPhy y MMAF). En el artículo muestran que esta estrategia integral supera a los clasificadores individuales, así surgió Meta-SVM/LR. Si bien las herramientas utilizadas para la predicción de Meta-SVM/LR están presentes en nuestro clasificador, cada una toma diferentes datos para generar sus modelos, por lo que no sería redundante utilizarla en nuestro set de entrenamiento.

Variant **E**ffect **S**coring **T**ools v**3** (VEST3) usa ~ 45.000 variantes patogénicas de la HGMD y ~ 45.000 variantes de alta frecuencia que se asumen neutrales encontradas en el Proyecto de Secuenciado del Exoma (Exome Sequencing Project). Utilizando únicamente datos de regiones codificantes entrenan un algoritmo de aprendizaje automático para predecir la patogenicidad de las variantes en el exoma humano[85].

FATHMM-MKL es una herramienta creada para evaluar la patogenicidad de variantes en todo el genoma[92]. Para ello usan variantes patogénicas y neutrales disponibles en HGMD anotadas con información de su conservación evolutiva (46-Way y 100-Way), modificación de histonas (ChIP-Seq), unión a factores de transcripción (TFBS Peak-Seq y SPP), compactación de la cromatina (DNase-Seq y FAIRE), contenido GC, segmentación del genoma y footprinting.

Fitness **Cons**equence (fitCons) divide su clasificación en 624 clases de elementos genómicos funcionales que derivan de agrupar datos de DNA-Seq, RNA-Seq y ChIP-Seq[93]. Éstos fueron usados para discriminar los sitios bajo selección en cada una de estas clases. A partir la importancia funcional de cada clase y la presión selectiva que esté actuando sobre dicha región se estima, para cada sitio, su rol en el fitness del individuo.

Exome **A**ggregation **C**onsortium (ExAC) es un proyecto público que agrupa la información nucleotídica del exóma de 60.706 individuos a nivel mundial[94]. Los alineamientos múltiples derivados permiten obtener la frecuencia poblacional para cada variante presente en el muestreo. Además de la frecuencia en el total de individuos, ExAC nos proporciona frecuencias especificas por población. Éstas están disponibles para

poblaciones africanas / afro-americanas, amerindias, asiáticas del este, finlandesas, europeas no-finlandesas, asiáticas del sur y otras.

Gen**ome** **A**ggregation **D**atabase (gnomAD) es una iniciativa conjunta que logró secuenciar el genoma entero de 15.708 y el exoma de 125.748 individuos no relacionados para la referencia hg19[95]. Al igual que con el caso anterior, esta nos proporciona información sobre la frecuencia general de las variantes y su frecuencia a nivel de poblaciones especificas. En el caso de gnomAD estas poblaciones son africanas / afro-americanas, amish, latinas, judías, asiáticas del este, finlandesas, europeas no-finlandesas, asiáticas del sur y otras. También podemos contar con las frecuencias específicas para hombres y mujeres.

Known **Vari**ants (Kaviar) es un consorcio que reúne 35 proyectos pudiendo contar con 64.600 exomas y 13.200 genomas de individuos no emparentados[96]. Kaviar excluye genomas de pacientes con cancer pero incluye algunos tipos de enfermedades particulares, sin concentrar sus esfuerzos en caracterizar una de ellas, el proyecto busca una caracterización general de la población humana. Esta base de datos proporciona información de la frecuencia general de las variantes, no las discrimina por poblaciones.

Genomic **E**volutionary **R**ate **P**rofiling (GERP) es una herramienta que usa máxima parsimonia para definir el efecto de la selección natural sobre cada sitio del genoma y con ello inferir su rol para el fitness del individuo[89]. Luego de definir las restricciones que impone la selección natural en cada sitio, GERP asigna un valor de "sustitución rechazada" (RS) que se define como el numero de sustituciones esperadas bajo neutralidad menos el numero de sustituciones encontradas en esa posición. Por lo que los valores positivos implican un déficit de substituciones por lo que el sitio estaría bajo restricción selectiva y por ende una mutación en dicha posición tendería a afectar más el fitness del individuo.

SiPhy es una herramienta que busca determinar la presión selectiva que sufre cada nucleótido en un genoma. Para ello calcula la conservación de cada base en un clado, como lo hacen la mayoría de los estimadores de conservación evolutiva. Sin embargo, SiPhy toma en cuenta también los patrones característicos de sustitución que vemos en regiones bajo presión selectiva[90].

PhyloP alinea el genoma completo de 29 especies de mamíferos para identificar regiones que estén bajo selección purificadora y funcionalmente enriquecidas[91]. PhyloP nos proporciona también el mismo estudio para un set de datos reducido a vertebrados llamado PhyloP7way_vertebrate.

Phylogenetic **A**nalysis with **S**pace/**T**ime models **Cons**ervation (phastCons) es una herramienta que busca identificar elementos conservados evolutivamente. Utiliza un modelo de cadenas de Markov escondidas (HMM) de dos estados, llamado phylo-HMM[97]. A partir de los alineamientos entrena dicho modelo por maxima verosimilitud y luego lo utiliza para predecir elementos conservados en el genoma. La estrategia difiere según la especie, por lo que el usuario debe modificar parámetros para adaptar el entrenamiento del modelo al organismo que se quiera estudiar.

Clinical **Inter**pretation for Genetic **Var**iants (InterVar) es una herramienta desarrollada por los mismos investigadores que crearon ANNOVAR, su objetivo es inferir la significancia clínica de las variantes nuevas basándose en 18 criterios[99]. Estos criterios buscan determinar el grado de patogenicidad de las variantes. Las variantes con probabilidad alta de ser benignas son asignadas al criterio BA1, mientras que las variantes con probabilidad alta de ser patogénicas son asignadas al criterio PVS1. Cada criterio toma diferentes propiedades biológicas para determinar el grado de patogenicidad de la variante, por lo que un mismo SNP puede ser asignado a mas de un criterio. Para anotar las variantes InterVar utiliza ANNOVAR. Una vez generado el perfil de criterios de cada SNP, la herramienta los usa para determinar si la variante es Benigna / Posiblemente benigna, Patogénica / Posiblemente patogénica o de Significado incierto.

Combined **A**nnotation-**D**ependent **D**epletion (CADD) es una aproximación integral a la anotación de variantes causales en análisis genéticos, particularmente variantes de penetrancia completa[88]. Para esto entrena una Maquina de Soporte de Vectores con kernel linear utilizando variantes simuladas y anotadas con más de 60

características genómicas. Los datos simulados incluyen variantes surgidas *de novo* (sin presión selectiva) y variantes fijadas en la población humana desde su divergencia con el chimpancé (mayormente neutras).

Deleterious **A**nnotation of Genetic Variants using **N**eural **N**etworks (DANN) tiene el mismo propósito y usa el mismo set de entrenamiento que CADD. Al usar un SVM con kernel linear, CADD no puede encontrar relaciones no-lineales entre las variantes. Buscando una aproximación complementaria, DANN entrena una red neuronal multicapa que es capaz de encontrar relaciones no-lineales. Esto le permite mejores predicciones si se cuenta con grandes cantidades de datos.

regSNPs es una estrategia que integra diferentes herramientas bioinformáticas para predecir la patogenicidad de una variante[27]. Tiene dos principales diferencias con el resto de estrategias, por un lado, calcula el efecto de la variante sobre la afinidad de unión al ADN y por otro, hace inferencias sobre los efectos fenotípicos que puede tener cada factor de transcripción sobre la región codificante.

Genome **W**ide **A**nnotation of **Va**riants (GWAVA) es una herramienta desarrollada específicamente para anotar variantes en regiones no-codificantes[100]. Ésta utiliza anotaciones disponibles en ENCODE/GENCODE (compactación de la cromatina, unión a factores de transcripción, modificación de histonas, etc), medidas de conservación evolutiva y contenido GC para entrenar tres algoritmos de Bosques Aleatorios (Random Forest) cada uno formado por 100 árboles.

Eigen es otra herramienta que implementa una aproximación integral para anotar la significancia clínica de las variantes en el genoma humano entero. A diferencia de los otros métodos disponibles, Eigen entrena un algoritmo no-supervisado de aprendizaje automático, por lo que no utiliza un set de entrenamiento etiquetado[101]. En esta aproximación Eigen hace uso de las predicciones generadas por los métodos de inferencia vistos anteriormente. Para cada una de estas anotaciones funcionales estima su precisión y luego las usa para generar una combinación lineal ponderada que ofrece una única anotación para cada variante.

Podemos ver que 9 de las 24 herramientas utilizadas (SIFT, PolyPhen2, Meta-SVM/LR, LRT, FATHMM, PROVEAN, VEST3, MutationTaster y MutationAssessor) utilizan las propiedades fisicoquímicas de los aminoácidos para inferir el efecto fenotípico causado por una mutación puntual. Además, el repositorio ExAC es generado a partir de exomas, por lo que informa la frecuencia de variantes en regiones codificantes. Y el predictor InterVar no presenta datos para gran parte de las variantes no-codificantes. En consecuencia, disponemos únicamente de 13 scores que hacen uso de otras características biológicas, permitiéndonos anotar NCV. Por un lado, Kaviar y gnomAD calculan la frecuencia de aparición de cada SNP en poblaciones humanas. Por otro lado, GERP, phyloP, phastCons y SiPhy determinan la presión selectiva y el grado de conservación de cada nucleótido para inferir su importancia funcional. Y por último, FATHMM-MKL, fitCons, regSNP, CADD, DANN, GWAVA y Eigen emplean aproximaciones integrales en las que generan modelos a partir de herramientas existentes y datos de compactación de la cromatina, unión a factores de transcripción, modificación de histonas, contenido GC y segmentación del genoma.

2. Imputación

Existen varios métodos para la imputación de datos faltantes. Cada uno genera una aproximación basada en algoritmos como los vecinos más cercanos (KNN) o Bosques Aleatorios (Random Forest). Estas herramientas deben ser capaces de inferir valores en varias columnas, no en una sola columna de resultado como suelen trabajar los algoritmos clásicos de AA. Por esta razón, no todos pueden ser usados para predecir valores ausentes. No existe, entre las herramientas desarrolladas para este propósito, una que sea ampliamente superior. Por estas razones decidimos evaluar el desempeño de diferentes métodos de imputación sobre nuestros datos buscando identificar el de mejor rendimiento.

Para ello tomamos los siguientes paquetes de R disponibles en **C**omprehensive **R** **A**rchive **N**etwork (CRAN): *DMwR* (v. 0.4.1), *Hmisc* (v. 4.2-0), *mice* (v. 3.6.0), *mi* (v0.10-2), *Amelia* (v. 1.7.5). Cada uno de éstos utiliza

su propia aproximación para imputar valores ausentes. La aproximación utilizada para evaluar el desempeño de algoritmos de imputación se presentará en el siguiente orden.

- Descripción de los algoritmos
- Métricas utilizadas para evaluar su desempeño
- Proceso por el cual se evalúa la imputación generada por los diferentes algoritmos.
- Proceso de evaluación de un método alternativo para imputar valores ausentes en una única columna.

La función *knnImputation* del paquete *DMwR*[102] de R busca completar los valores ausentes en un set de datos usando los *k* vecinos más cercanos (KNN). Este algoritmo posiciona el valor desconocido (NA) en un espacio multidimensional generando una media ponderada a partir de las variables que definen a sus *k* valores más similares. La cantidad de dimensiones del espacio están determinadas por la cantidad de variables con las que cuente nuestro set de datos.

La función *aregImpute* del paquete *Hmisc* de R busca imputar valores ausentes usando un modelo aditivo flexible. Para cada columna que tenga valores ausentes, *aregImpute* entrena dicho modelo de regresión no-paramétrica y lo usa para predecir tanto valores ausentes como valores observados[103]. Al predecir valores observados, la función puede definir medidas de precisión para su imputación, en este caso utilizando el coeficiente de determinación (R^2). Además, utiliza re-muestro con bootstrap para generar una distribución Bayesiana a partir de la cual pueden imputar el valor mas frecuente.

La función *mice* del paquete **M**ultivariate **I**mputation by **C**hained **E**quations (*MICE*) de R busca imputar valores ausentes usando cadenas de Markov por Monte Carlo (MCMC). Su método se basa en una "Fully Conditional Specification", un tipo de MCMC en el cual cada variable es imputada por separado usando su propio modelo[104]. Esto quiere decir que, a diferencia de lo visto en los métodos anteriores, no se toma en cuenta el resto de columnas de la tabla para imputar los valores ausentes de una variable.

La función *mi* del paquete *mi* de R busca imputar valores ausentes iterando modelos de regresión. Para ello genera una matriz (Y) con los valores ausentes y una matriz (X) con los valores observados. A través de un muestreo aleatorio de X completan todos los valores de Y. Luego utilizan ambas matrices para generar modelos de regresión a partir de los cuales imputar cada columna de Y. Este último proceso lo repiten hasta que los valores de Y converjan[105].

La función *amelia* del paquete *Amelia*[106] de R busca imputar valores ausentes usando un algoritmo de esperanza-maximización (EM). El abordaje de *amelia* consiste en crear una versión completa del set de datos haciendo bootstrap y a partir de ésta encuentra estimadores de máxima verosimilitud (parámetros) usando EM. Estos parámetros los usa luego para imputar los valores ausentes en el set de datos original. Este proceso lo repite "m" veces, donde los parámetros para imputar los valores del set de datos son estimados por EM en "m-1".

Para evaluar la precisión con la que cada método imputa los valores ausentes de una columna usamos dos medidas de error. Éstas son el Error Absoluto Medio (*MAE*) y la Raíz del Error Cuadrático Medio (*RMSE*). El Error Absoluto Medio toma la diferencia de cada valor predicho con el valor observado y devuelve el promedio de dicha diferencia. Mientras que la Raíz del Error Cuadrático Medio toma el cuadrado de la diferencia entre los valores y devuelve su raíz cuadrada. Por esta razón el *RMSE* penaliza los errores más grandes, mientras que el *MAE* penaliza por igual todos los errores reportando una medida aproximada del desempeño general del algoritmo. Usar estos dos parámetros en conjunto nos permite priorizar los métodos con mejor desempeño y al mismo tiempo evaluar la distribución de los valores imputados. A modo de ejemplo, el *MAE* puede reportar un error muy bajo para la imputación de una columna X utilizando un método A. Pero basándonos solo en este valor no podemos identificar sesgos en la predicción. Supongamos que la columna X tiene mayormente NAs MCAR y que el método A predice muy bien este tipo de NAs. Sin embargo, su desempeño es muy malo al predecir NAs MAR generando gran cantidad de valores fuera de los rangos esperados para X (outliers). Éstos

outliers pueden detectarse con el *RMSE* ya que penaliza fuertemente los errores mas severos. Así, al utilizar los parámetros *MAE* y *RMSE* en conjunto podemos priorizar los métodos que mejor predigan cada columna según la distribución de valores ausentes que tenga cada una de ellas.

Para estimar el error de cada método de imputación desarrollamos el siguiente proceso que itera por columna. En cada una se reemplazan aleatoriamente 600 valores observados por valores ausentes. El número de valores tomados se limitó a 600 con el objetivo de dejar suficientes valores observados disponibles para que el algoritmo pueda generar una inferencia robusta. Estos valores observados y sus índices son guardados en una variable X. Luego se procede a imputar todos los NAs del set de datos y se guardan, en una variable Y, los valores imputados en aquellos 600 índices reemplazados previo a la imputación. En este punto contamos con los valores observados (X) y los valores predichos (Y) de dicha columna. A partir de X e Y podemos calcular los errores (*MAE* y *RMSE*) del método que se está evaluando para la columna que se está imputando. Todos los valores predichos son luego reemplazados por sus valores originales guardados en la variable X y el proceso comienza de la misma manera para la columna siguiente hasta terminar el set de datos. Este proceso se repite idénticamente en cada método de imputación. De esta manera se obtienen los valores de *MAE* y *RMSE* para cada columna en cada método evaluado. Dado que la función *mi* requiere que se normalicen los datos antes de generar la imputación, el rango de valores sobre el que se calculan las medidas de error en este difiere del resto de los algoritmos. Para solucionarlo y poder comparar el desempeño de los métodos dividimos los valores de *MAE* y *RMSE* obtenidos para cada columna por el valor absoluto del rango de la misma. De esta manera obtenemos el *MAE* y *RMSE* de cada método y cada columna entre los valores de 0 y 1.

Cada una de las 15 columnas con las que cuenta nuestro set de datos tiene su propia distribución. Por ello es importante que el proceso itere por variable, de manera tal que se pueda estudiar el desempeño de cada algoritmo al imputar NAs en diferentes distribuciones. De esta manera podemos estar seguros que utilizamos el algoritmo más optimo en cada caso.

Antes de comparar el desempeño de los predictores filtramos aquellas que variantes que tuvieran en su mayoría datos faltantes (mas de 70%), ya que en esos casos la mayor información vendría de la imputación

La predicción de valores ausentes en la columna regsnp se realizó con un acercamiento diferente, ya que ninguno de los algoritmos estudiados infirió correctamente sus valores ausentes. Este acercamiento consiste en tomar regsnp como variable resultado, entrenar un modelo con las 15 columnas restantes (Significancia Clínica incluida) previamente imputadas y usar dicho modelo para predecir sus valores ausentes. A diferencia de los métodos evaluados anteriormente, éste se basa en un proceso de Aprendizaje Automático y no es capaz de imputar NAs en varias columnas.

Para evitar sesgos en nuestra predicción, balanceamos el set de entrenamiento pasando de 9.587 a 1.087 valores menores a 0.1. Dichos datos balanceados se utilizaron para entrenar un perceptrón de 7 capas con la función *mlp* del paquete *RSNNS* (v. 0.4-12) disponible en CRAN. Una vez entrenado, el modelo se utilizó para predecir los 1.292 valores ausentes en la anotación de la columna regsnp.

Una vez obtenidos los datos se evalúa la correlación de la matriz. Si existe una correlación considerable entre las variables, el clasificador puede sobrestimar la importancia de ciertas características biológicas. Si una misma característica biológica esta representada por varias columnas, el clasificador va a estar sesgado por la variación en dicho conjunto de columnas. Es decir, una característica biológica puede ser importante para la clasificación por estar sobre-representada en el set de entrenamiento y no por su verdadero peso en definir la patogenicidad de una variante. Para evitar este sesgo medimos la correlación de *pearson* que existe entre cada una de las columnas de nuestro set de datos. Esto se realizó con la función *cor* del paquete *stats* (v. 3.6.1) R especificando la opción *method = "pearson"*. Para definir que dos variables estaban correlacionadas tomamos un umbral de 0.80 según *pearson*.

Una vez filtradas las columnas correlacionadas, se obtiene un set de 11 columnas a utilizar para entrenar diferentes algoritmos.

3. Entrenamiento

Para el paso de entrenamiento disponemos de un gran número algoritmos que buscan modelar la estructura de los datos haciendo uso de diferentes propiedades estadísticas. Dentro del aprendizaje supervisado se tiende a diferenciar los algoritmos según su interpretabilidad y su precisión (Figura 1). En métodos como los arboles de decisión es relativamente sencillo identificar las transformaciones matemáticas aplicadas para generar el modelo, sin embargo, su precisión suele ser limitada. Por el contrario, los métodos basados en redes neuronales se caracterizan por generar modelos muy precisos al costo de una menor interpretabilidad. La baja interpretabilidad depende de la multidimensionalidad de la red neuronal en la que se computa el proceso. Cada capa de neuronas genera múltiples transformaciones por lo que se vuelve difícil restablecer el camino de sinapsis que siguió la señal desde la capa de salida hasta la capa inicial de neuronas. El proceso mas utilizado en aprendizaje automático consiste en comenzar entrenando modelos altamente interpretables y avanzar gradualmente hacia modelos más precisos. De esta manera se puede identificar las variables con mayor peso a la hora de generar el modelo. Además, de generarse un modelo preciso a partir de un algoritmo interpretable no sería necesario recurrir a un modelo de redes neuronales en el que es difícil conocer los pesos internos de cada variable.

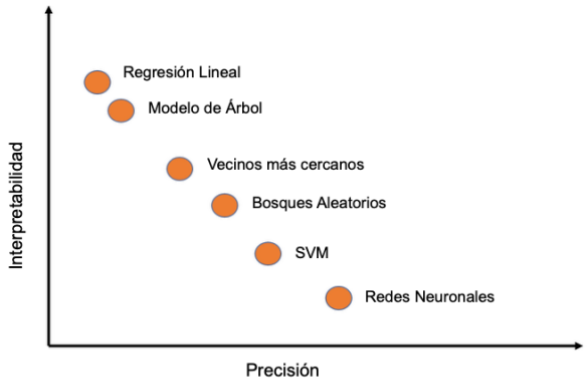


Figura 1. Representación de la interpretabilidad de 6 métodos de clasificación según un estimado de su precisión.

Siguiendo las buenas prácticas en Aprendizaje Automático, comenzamos entrenando un modelo de regresión lineal como línea base por su simplicidad y ser fácilmente interpretable. Estos modelos buscan establecer una función lineal que permita dividir la variable resultado a partir de las variables de entrenamiento. La distancia de cada punto a la línea es proporcional a su posición en la variable resultado predicha, por lo que la predicción es de carácter cuantitativo (regresión). Al depender de una función lineal, es útil para modelar datos fáciles de separar (ej. mujer/hombre ~ altura) y entender las transformaciones realizadas (interpretabilidad). Sin embargo, falla en modelar casos mas complejos donde la variable resultado no se puede separar en dos grupos claros usando un solo plano dimensional.

Luego pasamos a entrenar modelos de árbol. Los modelos basados en arboles de decisión pueden generar tanto regresiones como clasificaciones, por lo que su resultado puede ser tanto cuantitativo como cualitativo. Se basan en un proceso jerárquico de decisiones, cada una con una salida intermedia asociada que se representa en nodos[107]. De cada nodo derivan aristas que representan la decisión por la cual se llega al siguiente nodo, y el proceso continúa. En cada paso se evalúa la decisión tomada usando división binaria recursiva (clasificación) o la suma residual de los cuadrados (regresión). El único nodo que no deriva de una arista es el nodo inicial,

por lo que hay una sola manera de llegar a los datos de entrada. Esta característica hace fácil interpretar las transformaciones llevadas a cabo por el proceso de decisión. Además, el eje sobre el que se dividen los datos varía de un nodo al otro, por lo que es un algoritmo útil para modelar problemas no-lineales.

El siguiente método entrenado se conoce como los *k* vecinos mas cercanos (k-NN). Éste consiste en posicionar cada entrada en un plano Euclidiano en el que cada variable define un eje del espacio multidimensional. k-NN funciona bajo la suposición de que las entradas similares van a encontrarse próximas en este espacio. Se trata de un algoritmo no-paramétrico muy versátil por lo que se usa tanto para aprendizaje supervisado como no-supervisado (clustering) e imputación de valores ausentes, pudiendo entrenar regresiones y clasificaciones. El único valor a ajustar (*k*) define el tamaño del vecindario que se toma para establecer los valores similares a un punto. Si el vecindario es muy chico la predicción va a ser similar a los datos originales (overfitting), mientras que para vecindarios muy grandes la predicción va a tener muy poca varianza[36].

Los algoritmos de Bosques Aleatorios (RF) son básicamente un conjunto de árboles de decisión entrenados independientemente en paralelo cuya salida se promedia (regresión) o vota (clasificación) para obtener un único reporte. Fueron creados por Breiman & Cutler[108] con el objetivo de reducir el sobre-entrenamiento que sufrían los arboles de decisión convencionales[109]. Para ello hacen uso de la agregación por bootstrap y la selección de variables al azar, proceso por el cual se controla la varianza de los arboles. Éste consiste en mantener *m* variables en cada nodo a partir de las cuales se toma la decisión, las *m* variables tomadas varia aleatoriamente entre los nodos. Al entrenar cada árbol sobre un set reducido de variables, la predicción final no esta sobre-ajustada al set de entrenamiento. Este parámetro ajustable se conoce como número de variables disponibles para dividir o *mtry*. El proceso se repite para el crecimiento de todos los arboles del bosque. La cantidad de arboles a crecer se ajusta con otro parámetro (*ntree*).

Los modelos Gradient Tree (GBT) son otro derivado de los arboles de decisión. A diferencia de los Bosques Aleatorios que utilizan bagging, el Boosted Tree computa múltiples árboles de decisión concatenados de forma tal que la salida de uno ayuda a reducir el error del siguiente. Esto se consigue tomando los residuales del árbol anterior para crecer el siguiente árbol. Si bien este proceso aumenta la precisión del modelo, disminuye su interpretabilidad con respecto a los Bosques Aleatorios y lo hace mas propenso al sobre-ajuste. Para regularlo se utilizan un gran numero de hiperparámetros como el número de iteraciones o arboles (*nrounds*), la profundidad máxima que puede crecer cada árbol (*max_depth*), la tasa de reducción o poda de los arboles (*eta*), la reducción mínima (*gamma*), porcentaje de columnas a dividir (*colsample_bytree*), entre otros. En el caso de los Boosted Trees también podemos entrenar tanto modelos de regresión como modelos de clasificación.

Aumentando la precisión del algoritmo encontramos las Maquinas de Soporte de Vectores (SVM). Éstas mapean sus puntos a un espacio multidimensional y buscan computar los vectores que más separen los datos. Su principal característica es que, al manejarse con vectores, puede entrenar modelos no-lineales transformando los mismos según diferentes tipos de kernel. Éstos pueden ser kernels lineales, polynomiales o radiales. Según el kernel que se utilice es cómo se va a deformar el hiperplano y por ende el tipo de datos que puede modelar. Al igual que los otros métodos, éste permite generar modelos de regresión y modelos de clasificación. La dificultad en interpretarlos se deriva principalmente del uso de kernels, esto hace difícil identificar los datos de entrada a partir de la predicción generada. El principal parámetro a ajustar al entrenar un SVM es el uso del kernel con sus parámetros derivados, como puede ser el grado de un kernel polinomial o sigma de un kernel radial. Pero también se pueden ajustar parámetros como el soft margin (*C*) y la regularización (*tau*).

Las redes neuronales son el algoritmo más preciso y menos interpretable por excelencia. Están formadas de al menos tres capas de nodos (neuronas) conectadas por aristas (sinápsis), estas son una de entrada, una de salida y un número variable de capas ocultas (intermedias). La analogía al sistema nervioso proviene del umbral de activación que debe atravesar una neurona para transmitir su información a la capa siguiente. Dicho umbral es determinado por la función de activación, una transformación realizada en la primera iteración del proceso de aprendizaje que tiene como objetivo introducir no-linearidad a la red. Además de dicha función hay otros

parámetros que se pueden ajustar para definir la arquitectura de nuestra red. La estructura más favorable va a depender del set de datos que queríamos modelar. Los otros parámetros son el número de capas intermedias, el número de neuronas por capa, el número de veces que cada dato atraviesa la red (iteraciones) y el ratio de aprendizaje. Este último determina la importancia que va a tener el resultado de cada iteración sobre el ajuste de los pesos de cada neurona para la siguiente iteración en un proceso llamado propagación hacia atrás o *back propagation*. El entrenamiento comienza asignando pesos aleatorios a cada neurona y cada sinápsis, lo cual produce un resultado aleatorio. A partir de este resultado se ajustan, desde la capa de salida hacia las capas intermedias, los nuevos pesos usando la propagación hacia atrás. Este proceso se repite para la cantidad de iteraciones deseadas o hasta que el ratio de aprendizaje se estabilice. La precisión de estos algoritmos escala exponencialmente al entrenarlos con grandes cantidades de datos, pero su interpretabilidad decrece a medida que complejizamos la arquitectura de la red. Por su precisión son métodos muy útiles y llamativos, pero requieren trabajo posterior para entender la predicción que esta generando. Una buena forma de hacerlo es entrenar gradualmente algoritmos más interpretables como ya hemos mencionado.

Para el entrenamiento se evaluaron tres acercamientos (o escenarios) diferentes, cada uno con su propio set de entrenamiento. El objetivo de esta comparación es determinar el acercamiento mas informativo para separar variantes patogénicas de variantes benignas.

i) El primer acercamiento consta de dos pasos. En primero lugar identificar y filtrar variantes de significado incierto. Y en segundo lugar separar variantes benignas de variantes patogénicas. Dado que la amplia mayoría de las variantes no codificantes están anotadas como Significado incierto, este clasificador busca priorizar variantes con características de significancia dentro de las regiones no codificantes. Para ello generamos un set de entrenamiento con resultado binario, tomando 2.500 variantes con significancia, de las cuales 1.250 son patogénicas y 1.250 son benignas y 2.500 variantes de significado incierto (Figura 4. C). De aquí en adelante vamos a referirnos a esta distribución del set de entrenamiento cómo el escenario de Significancia incierta - Significancia cierta (SI-SC).

ii) El segundo acercamiento tenía cómo objetivo evaluar directamente la patogenicidad de las variantes en diferentes niveles. Para ello generamos un set de entrenamiento de seis clases donde las variantes están anotadas según su patogenicidad en orden creciente. Se tomaron 2.000 variantes benignas, 2.000 posiblemente benignas, 200 de significado incierto / benignas, 2.000 de significado incierto, 20 de significado incierto / patogénicas, 2.000 posiblemente patogénicas y 2.000 patogénicas (Figura 4. B). De aquí en adelante vamos a referirnos a esta distribución del set de entrenamiento cómo el escenario de Todas las clases o Todas las categorías (TC).

iii) La última aproximación tenía como objetivo clasificar variantes patogénicas o benignas de manera binaria. En este caso, se puede obtener una probabilidad de pertenencia a cada clase. Para este paso generamos un set de entrenamiento binario en el que tomamos 4.000 variantes benignas y 4.000 variantes patogénicas (Figura 4. A). De aquí en adelante vamos a referirnos a esta distribución del set de entrenamiento cómo el escenario Benignas - Patogénicas (B-P).

Los algoritmos fueron entrenados con las mismas funciones para cada uno de los tres casos. Esto implica que se entrenaron una o varias regresiones lineales, árboles de decisión, k-NNs, Bosques Aleatorios, Gradient Boosted Trees, SVMs y Redes Neuronales para cada uno de los tres sets de entrenamiento generados. Las funciones utilizadas en cada caso fueron las siguientes.

- *lm* del paquete *caret* (v. 6.0-84) de R para las regresiones lineales.
- *treebag* del paquete *caret* (v. 6.0-84) de R y *J48* del paquete *RWeka* (v. 0.4-41) de R para los árboles de decisión.
- *knn* del paquete *caret* (v. 6.0-84) de R para el método de vecinos más cercanos.
- *randomForest* del paquete *randomForest* (v. 4.6-14) de R para los Bosques Aleatorios.

- *xgboost* (opción *booster = "gbtree"*) del paquete *xgboost* (v. 0.90.0.2) de R para los Gradient Boosted Trees.
- *ksvm* (opción *kernel = "rbfdot"* o *"polydot"*) del paquete *kernlab* (v. 0.9-29) de R para las Maquinas de Soporte de Vectores.
- *MLPClassifier* del modulo *neural_network* de la librería *sklearn* (v. 0.20.4) de Python para las Redes Neuronales.

Los parámetros utilizados para entrenar cada clasificador en cada uno de los tres casos se muestran en el anexo (Tabla suplementaria 1).

3. Evaluación

El análisis del desempeño de los clasificadores se puede dividir en dos etapas, éstas son la validación interna y la evaluación. Para ello el primer paso es dividir nuestro set de datos en dos subconjuntos, uno será el set de entrenamiento sobre el que se va a realizar la validación interna y otro será el set de evaluación.

El objetivo de la validación interna es encontrar los hiperparámetros óptimos para entrenar el algoritmo. Para ello, uno de los métodos más usados es la validación cruzada en *k* conjuntos (*k-fold cross-validation*). Ésta consiste en dividir aleatoriamente el set de entrenamiento en *k* subconjuntos, usar *k-1* subconjuntos para entrenar el modelo con ciertos parámetros y el restante para calcular las métricas de error (MAE, RMSE, Matriz de Confusión, entre otras) obtenidas con dichos parámetros. Este proceso se repite *k* veces cambiando el subconjunto utilizado para validar y alternando los parámetros. Una vez terminado el proceso obtenemos los hiperparámetros mejor ajustados para entrenar el modelo y métricas de error obtenidas con dicho ajuste. En nuestro caso, se utilizaron 10 subconjuntos para realizar la validación cruzada (10-fold cross validation). La métrica utilizada para comparar el desempeño de los modelos con cada set de hiperparámetros es la Precisión general o "Accuray", es decir el porcentaje de variantes correctamente clasificadas tanto para la clase positiva como para la clase negativa. Así, el set de hiperparámetros seleccionado para entrenar el modelo será aquel que produzca la mayor Precisión general sobre el set de validación interno.

Por otro lado, la evaluación se realiza sobre los datos que fueron excluidos del set de entrenamiento en el primer paso o set de evaluación. Estos datos no fueron utilizados para generar el modelo en ninguna de sus etapas. Por lo que no formaron parte del escalado de los datos, la selección de hiperparámetros por validación cruzada o del entrenamiento del modelo. El objetivo de este paso es evaluar el desempeño del clasificador sobre datos ajenos al entrenamiento y estimar así el sobre-ajuste del mismo.

Las métricas con las que se evalúa cada método son diferentes para clasificaciones y regresiones. Es decir, depende de si la predicción es categórica (cualitativa) o continua (cuantitativa). En nuestro caso, el único caso continuo son las regresiones lineales entrenadas con la función *lm* del paquete *caret*. El resto de los modelos entrenados con clasificaciones, por lo que su predicción es categórica.

Para evaluar el rendimiento de una regresión se calculan métricas de error continuas como el Error Absoluto Medio (*MAE*) y la Raíz del Error Cuadrático Medio (*RMSE*) detalladas anteriormente.

Por otro lado, las clasificaciones se evalúan usando una Matriz de Confusión y el área debajo de la curva ROC (Característica Operativa del Receptor). Las Matrices de Confusión se generan para estudiar la cantidad de falsos positivos, y falsos negativos asignados a cada clase. Para ello, se disponen en columnas las clases de los valores predichos y en filas las clases de los valores observados (Figura 5). Vamos a tener tantas columnas y tantas filas como clases haya en nuestra clasificación. Éstas suelen acompañarse de las medidas de *Sensibilidad* y *Especificidad*. La *Sensibilidad* se calcula como el cociente entre los verdaderos positivos y la suma de verdaderos positivos y falsos negativos (VP / (VP+FN)). Es decir, representa el desempeño del clasificador en asignar bien las entradas que pertenecen a cada categoría. Mientras que la *Especificidad* se calcula como el

cociente entre los verdaderos negativos y la suma de verdaderos negativos y falsos positivos (VN / (VN+FN)). Es decir, representa el desempeño del clasificador en no asignar a una categoría las entradas que no pertenecen a la misma.

Por su parte, la curva ROC se genera a partir de la relación que existe entre la *Sensibilidad* y la *Especificidad* del clasificador. En esta, el *x* se define cómo *1-Especificidad* o el ratio de falsos positivos e *y* cómo la *Sensibilidad*. En un caso completamente aleatorio esperamos obtener un ratio de verdaderos positivos muy bajo, por lo que una *Sensibilidad* cercana a *1-Especificidad*. Esto genera para las predicciones obtenidas por azar una diagonal que atraviesa el gráfico de izquierda a derecha (Figura 6). En el otro extremo, una clasificación perfecta no produciría ningún falso negativo, solo verdaderos positivos, por lo que su curva ROC tendería a valores de *y* (*Sensibilidad*) cercanos a 1 para valores de *x* (*1-Especificidad*) próximos a 0. Así se produce un área entre la curva ROC y la diagonal generada por azar que crece con el desempeño del clasificador. El área debajo de la curva ROC o AUROC es un método muy usado para comparar el desempeño general de dos clasificadores. Aunque no es el único, existen otros como el valor F o el Área Debajo de la Curva de Precisión-Recall (AUPRC).

Para la evaluación de las regresiones del presente trabajo se utilizaron las funciones *mae* y *rmse* del paquete *Metrics* (v. 0.1.4) de R. Para obtener las Matrices de Confusión, así como las medidas de *Sensibilidad* y *Especificidad* de los clasificadores entrenados se utilizó la función *confusionMatrix* del paquete *caret* (v. 6.0-84) de R. Para generar sus respectivas curvas ROC y obtener el área debajo de la curva se utilizo la función *roc* del paquete *pROC* (v. 1.15.3) de R.

Se evaluaron únicamente los modelos entrenados en el escenario B-P, para ello se utilizaron dos sets de evaluación. Uno se compone de 421 variantes Patogénicas, 4.075 Posiblemente Patogénicas, 3.170 variantes de Significado Incierto, 4.415 Posiblemente Benignas y 1.092 variantes Benignas (Figura 4. A). El otro de variantes anotadas en un principio como Significado Incierto y luego re-anotadas como Benignas o Patogénicas (ver Métodos 1.), estas son 493 variantes de Significado incierto / Benignas y 43 variantes de Significado Incierto / Patogénicas (Tabla 1). Este último tiene el objetivo de evaluar el desempeño de dicho escenario en clasificar variantes de Significado Incierto.

5. Predicción en genomas uruguayos

Para varios de los pacientes del proyecto URUGENOMES no fue posible conseguir un diagnóstico molecular de su enfermedad al estudiar sus regiones codificantes, UTRs y sitios de splicing próximos al exón, por lo que era de interés estudiar las variantes no codificantes. Para ello se utilizó el clasificador RF entrenado con el escenario B-P para priorizar variantes en las regiones no codificantes de dichos pacientes.

Primeramente, para poder aplicar el modelo entrenados al genoma de un paciente uruguayo se anotaron los datos (vcf) con las mismas bases de datos que el set de entrenamiento. Para ello se anotan las variantes no-codificantes utilizando ANNOVAR y se filtran aquellas con más de 70% NAs. Los valores ausentes se imputan utilizando la metodología optimizada para el set de entrenamiento. Las variantes priorizadas en dicho set (ver Resultados) se entregaron a un genetista clínico para que las evalúe en detalle en el contexto de la historia clínica del paciente. Por restricciones de ética y privacidad solo el genetista puede conocer las características fenotípicas que presenta el paciente por lo que a los efectos de este trabajo de grado no se pueden realizar análisis posteriores con dicho set de variantes.

Resultados

Imputación:

De las 97.736 variantes no-codificantes anotadas de ClinVar, 31.218 no presentan información y 76.563 presentan menos de 9 anotaciones (Tabla 2). Para mantener el carácter biológico de las anotaciones

conservamos únicamente las 21.173 variantes con al menos 9 anotaciones reales. Dicho set de datos con buen porcentaje de anotación en variantes no-codificantes presenta un 39.1% de valores ausentes (Figura 2, A) para sus 15 bases de datos. Mientras que, a modo de ejemplo, las 282.249 variantes codificantes obtenidas de ClinVar presentan únicamente 9.5% de valores ausentes (Figura 2, B) en sus 22 bases de datos.

Porcentaje de NAs (%)	0	6.2	12.5	18.7	25.0	31.2	37.5	43.7	62.5	68.7	75.0	81.2	87.5	93.7	100
N° de variantes	245	407	680	1.165	1.106	4.938	1.114	31	11.487	6.081	19.611	6.247	9.735	3.671	31.218

Tabla 2. Número de variantes en regiones no-codificantes según su porcentaje de valores ausentes.

El desempeño de cada imputador se evaluó con el proceso detallado anteriormente (ver Métodos). De esa manera obtuvimos las métricas de error (*MAE* y *RMSE*) al imputar NAs con cada algoritmo en cada columna (Tabla 3). Lo primero a destacar es que no existe un método que supere al resto en las 15 variables estudiadas. Esto refuerza la importancia de evaluar el desempeño de cada imputador iterando sobre variables, teniendo en cuenta así la distribución de valores ausentes propia en cada una.

Por otro lado, vemos que todos los predictores superan la calidad de la imputación por el promedio (Tabla 3). Como era de esperar, reemplazar los valores ausentes por el promedio de la variable no sigue las características biológicas de cada variante, produciendo un set de datos poco informativo.

En amarillo (Tabla 3) se muestra el algoritmo con mejor desempeño para cada una de las variables (filas). En principio, el algoritmo basado en vecinos más cercanos (k-NN) parece superar al resto. Para los casos CADD, DANN, fathmm, GERP, phyloP7, phastCons7, phastCons20, SiPhy, GWAVA y Eigen, la función *knnImputation* predice valores más próximos al valor real de la columna. Esto se traduce en errores más pequeños al evaluar tanto *MAE* como *RMSE*. En los casos particulares de fitCons y gnomAD el algoritmo basado en k-NN produjo imputaciones con un Error Absoluto Medio menor al resto de los métodos, pero la Raíz Cuadrada del Error Cuadrático Medio fue superada por otro imputador en ambos casos. El predictor integral de patogenicidad (fitCons) presentó un *RMSE* 0.01 puntos menor al imputar sus NAs con la función *mi*. Mientras que la base de datos de frecuencias (gnomAD) presentó un *RMSE* 0.02 puntos menor al imputar sus NAs con la función *amelia*. Si bien la predicción generada por *knnImputation* mostró errores más abultados para los dos casos mencionados anteriormente, la diferencia es muy pequeña y el Error Absoluto Medio muestra una mejor imputación por parte del algoritmo basado en vecinos más cercanos. Por ende, para las 12 variables mencionadas hasta ahora se decidió utilizar *knnImputation* como algoritmo para predecir sus valores ausentes.

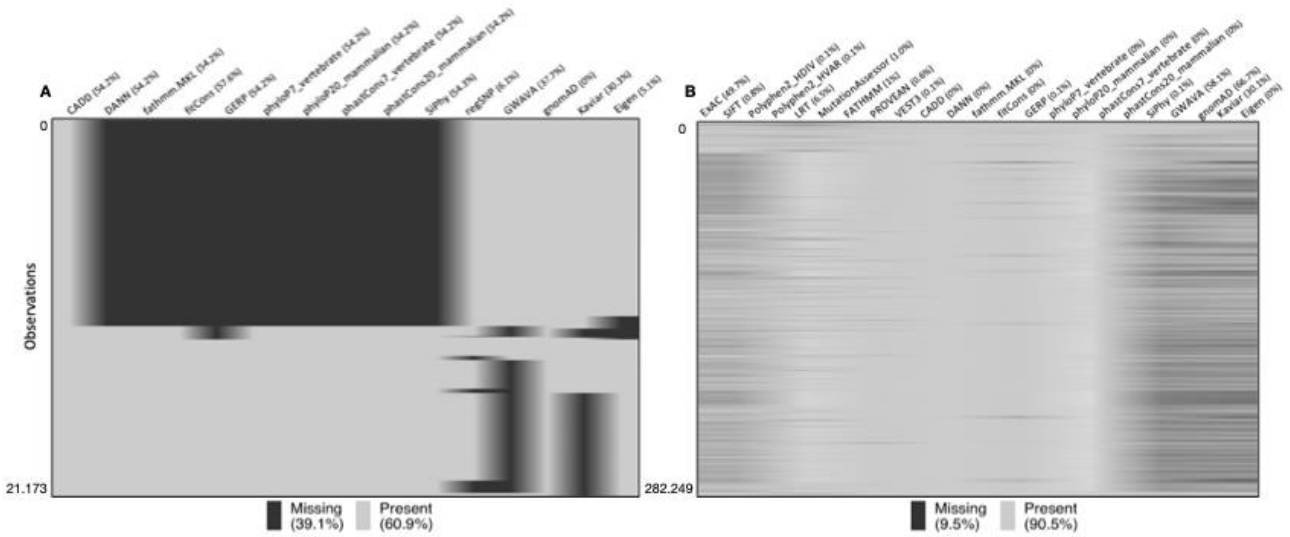


Figura 2. A. Patrón de valores ausentes en variantes no-codificantes. **B.** Patrón de valores ausentes en variantes codificantes.

En los casos de phyloP20 y Kaviar se observa que el algoritmo *mi* genera predicciones mas precisas que el resto de los imputadores, al evaluar tanto *MAE* como *RMSE* (Tabla 3). En el caso de Kaviar, su Error Absoluto Medio es ligeramente menor al obtenido con *knnImputation*, sin embargo, vemos un *RMSE* 0.11 puntos menor a la predicción generada por k-NN. Para la variable de conservación evolutiva (phyloP20) la diferencia es aun mas drástica, mi predice los NAs de esta columna con un *MAE* 0.06 puntos menor y un *RMSE* 0.15 puntos menor a *knnImputation*. Por las métricas de error obtenidas se optó por el algoritmo *mi* para imputar los valores ausentes de las columnas phyloP20 y Kaviar.

Ninguno de los 6 métodos evaluados proporcionó una imputación confiable para la columna regsnp, por ello decidimos imputar sus valores ausentes entrenando una red neuronal del tipo MLP, las cuales suelen generar muy buenas predicciones. Los valores reales de regsnp se encuentran muy sesgados hacia 0 lo cual se soluciono balanceando el set de entrenamiento (Figura 3). La aproximación utilizada parece ser exitosa ya que redujo el Error Absoluto Medio a 0.06 y la Raíz Cuadrada del Error Cuadrático Medio a 0.08 (Tabla 3). Esto se traduce en un 6% y 8% de error respectivamente. Estos valores están dentro de los errores aceptados para las otras variables. Así, al predecir los valores ausentes de regsnp usando un modelo de Perceptrón Multicapa, podemos mantener lo más posible el carácter biológico de nuestras anotaciones.

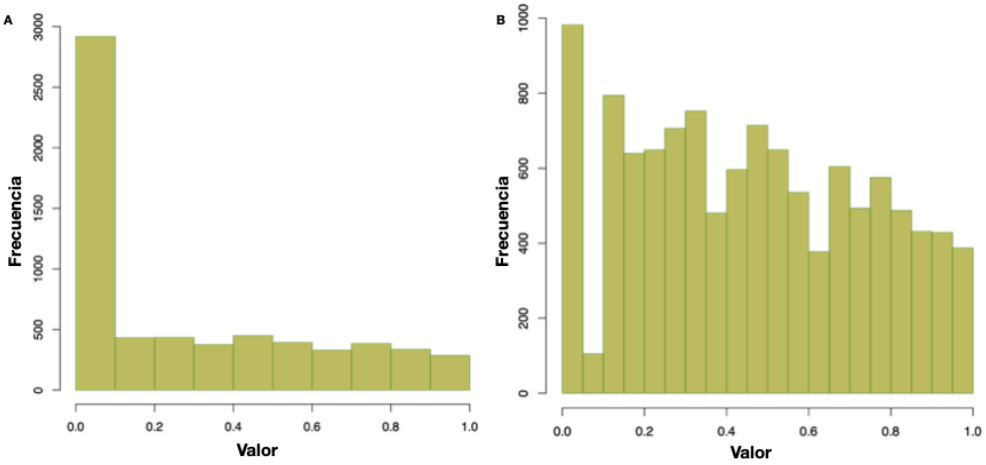


Figura 3. A. Histograma mostrando la distribución de valores presentes en la columna regsnp. **B.** Histograma mostrando la distribución de valores de regsnp presentes en el set de entrenamiento del perceptrón multicapa.

Columna	Rango	KNN		Hmisc		MICE		Amelia		mi		Promedio		Red Neural	
CADD	0 - 40	0.03	0.05	0.06	0.10	0.05	0.06	0.04	0.06	0.03	0.05	0.22	0.25		
DANN	0 - 1	0.01	0.04	0.01	0.04	0.04	0.05	0.04	0.05	0.04	0.06	0.12	0.19		
fathmm	0 - 1	0.04	0.09	0.08	0.13	0.09	0.13	0.09	0.13	0.07	0.10	0.32	0.36		
fitCons	0-0.834	0.12	0.18	0.18	0.22	0.15	0.20	0.16	0.21	0.13	0.17	0.13	0.17		
GERP	-11 - 6	0.02	0.04	0.04	0.06	0.04	0.06	0.05	0.06	0.05	0.07	0.15	0.16		
phyloP7	-4 - 1	0.01	0.04	0.06	0.14	0.04	0.05	0.04	0.05	0.04	0.05	0.41	0.55		
phyloP20	0 - 1	0.10	0.21	0.25	0.62	0.19	0.27	0.18	0.26	0.04	0.06	0.51	0.61		
phastC7	0 - 1	0.13	0.19	0.16	0.25	0.20	0.26	0.20	0.26	0.21	0.27	0.58	0.71		
phastC20	0 - 1	0.12	0.19	0.14	0.23	0.20	0.26	0.21	0.27	0.29	0.36	0.32	0.37		
SiPhy	0 - 20	0.08	0.11	0.08	0.11	0.13	0.17	0.13	0.17	0.13	0.17	0.16	0.19		
regsnp	0 - 1	0.23	0.36	0.20	0.30	0.26	0.35	0.24	0.35	0.23	0.31	0.44	0.52	0.06	0.08
GWAVA	0 - 1	0.11	0.13	0.16	0.12	0.145	0.186	0.146	0.182	0.18	0.22	0.11	0.14		
gnomAD	0 - 1	0.02	0.08	0.17	0.31	0.03	0.07	0.03	0.06	0.17	0.21	0.28	0.33		
Kaviar	0 - 1	0.05	0.16	0.19	0.38	0.04	0.09	0.07	0.10	0.04	0.05	0.25	0.39		
Eigen	-2.4 - 3.2	0.03	0.05	0.16	0.28	0.06	0.09	0.07	0.09	0.07	0.11	0.22	0.25		
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE

Tabla 3. Precisión (MAE y RMSE) de la imputación de cada método sobre datos de cada columna. En amarillo (🟡) se muestra el método con mejor desempeño en cada caso.

En resumen, para imputar los valores ausentes de las columnas CADD, DANN, fathmm, fitCons, GERP, phyloP7, phasCons7, phastCons20, SiPhy, GWAVA, gnomAD y Eigen se utilizó la predicción generada por la función *knnImputation* del paquete *DMwR* basada en vecinos más cercanos. Para imputar los valores ausentes de las columnas phyloP20 y Kaviar se utilizó la predicción generada por la función *mi* del paquete *mi* de R. Y por último, para predecir los valores ausentes de la columna regsnp se generó un modelo de perceptrón multicapa entrenado sobre nuestro propio set de datos imputados (14 variables + Significancia Clínica) a partir de la función *mlp* del paquete *RSNNS* de R.

Una vez finalizada la imputación se obtuvo un set de 21.173 variantes con anotaciones vinculadas con su carácter biológico, lo cual puede ayudar a definir patrones distinguibles entre SNPs patogénicos y benignos.

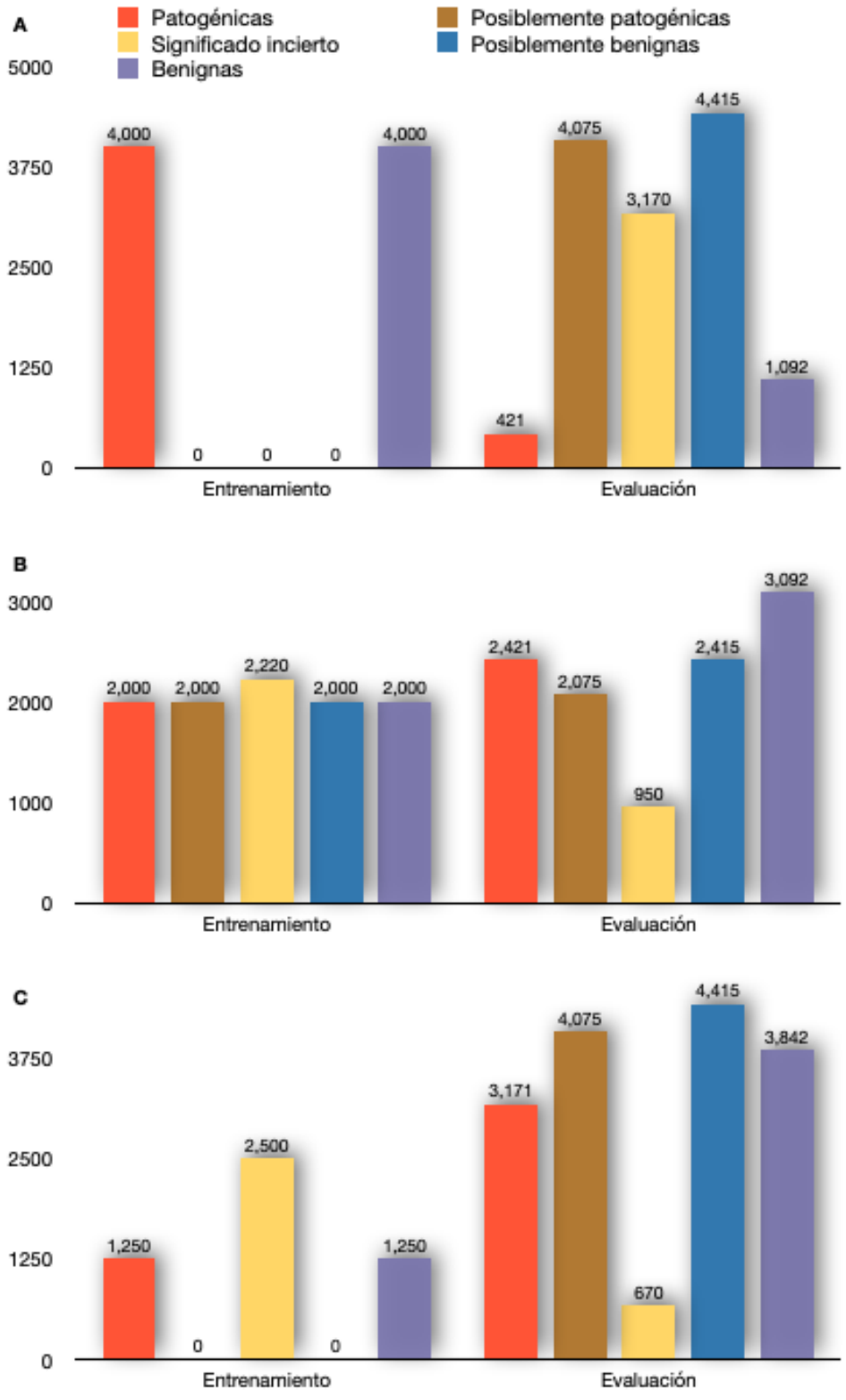


Figura 4. Distribución por clases de los sets de entrenamiento (izquierda) y evaluación (derecha) para la identificación de variantes Benignas - Patogénicas, Todas las clases y Significancia incierta - Significancia cierta (A, B y C respectivamente).

Preparación de datos:

Podemos ver que los predictores integrales (CADD, DANN, fathmm y Eigen) están altamente correlacionados con las medidas de conservación evolutiva (GERP, phyloP, phastCons, SiPhy y Eigen) (Tabla 4. A). Esto se

debe a que los predictores integrales toman la conservación de las bases entre las variables usadas para generar su predicción. De mantener todas estas variables estaríamos sobre-estimando el peso de la conservación evolutiva en definir una variante patogénica. Por ello descartamos las columnas CADD, phyloP7, phastCons7 y SiPhy.

La primera (CADD) es la que presenta más correlación, tanto con otros predictores integrales como con las medidas de conservación (Tabla 4. A). En cuanto a las bases de datos phyloP7 y phastCons7, éstas fueron generadas con alineamientos a partir de 7 especies de mamíferos. Mientras que phyloP20 y phastCons20 se generaron usando 20 especies de vertebrados. Al haberse generado a partir de un número mayor de especies, los sitios conservados en dicho alineamiento son mas relevantes funcionalmente que los sitios conservados en el alineamiento de 7 especies. Adicionalmente, las dos primeras presentan más correlación con los predictores de patogenicidad que las últimas dos. Por ambas razones decidimos descartar las columnas phyloP7 y phastCons7 de nuestro set de datos. Por último, la columna SiPhy fue descartada por estar muy correlacionada con los tres métodos restantes de predicción, particularmente DANN y fathmm.

De esta forma se intento minimizar la sobrestimación de alguna propiedad biológica particular. Las únicas variables altamente correlacionadas son los predictores integrales de patogenicidad entre sí (Tabla 4. B).

A

	CADD	DANN	fathmm	fitCons	GERP	phyloP7	phyloP20	phastC7	phastC20	SiPhy	regsnp	GWAVA	gnomAD	Kaviar	Eigen
CADD															
DANN	0.95														
fathmm	0.96	0.94													
fitCons	-0.65	-0.62	-0.65												
GERP	0.94	0.92	0.94	-0.66											
phyloP7	0.84	0.88	0.86	-0.61	0.86										
phyloP20	0.78	0.76	0.78	-0.56	0.82	0.73									
phastC7	0.84	0.80	0.84	-0.54	0.81	0.74	0.67								
phastC20	0.84	0.79	0.82	-0.51	0.78	0.70	0.66	0.89							
SiPhy	0.91	0.87	0.90	-0.60	0.89	0.80	0.70	0.82	0.80						
regsnp	-0.62	-0.59	-0.63	0.48	-0.59	-0.54	-0.47	-0.57	-0.57	-0.61					
GWAVA	0.66	0.65	0.65	-0.63	0.69	0.59	0.49	0.65	0.63	0.69	-0.51				
gnomAD	-0.34	-0.45	-0.37	0.23	-0.33	-0.64	-0.16	-0.31	-0.30	-0.37	0.25	-0.22			
Kaviar	-0.28	-0.36	-0.31	0.19	-0.27	-0.50	-0.13	-0.25	-0.26	-0.31	0.21	-0.18	0.77		
Eigen	0.90	0.85	0.89	-0.63	0.89	0.77	0.78	0.81	0.81	0.86	-0.59	0.61	-0.27	-0.22	

B

	DANN	fathmm	fitCons	GERP	phyloP20	phastC20	regsnp	GWAVA	gnomAD	Kaviar	Eigen
DANN											
fathmm	0.94										
fitCons	-0.62	-0.65									
GERP	0.92	0.94	-0.66								
phyloP20	0.76	0.78	-0.56	0.82							
phastC20	0.79	0.82	-0.51	0.78	0.66						
regsnp	-0.59	-0.63	0.48	-0.59	-0.47	-0.57					
GWAVA	0.65	0.65	-0.63	0.69	0.49	0.63	-0.51				
gnomAD	-0.45	-0.37	0.23	-0.33	-0.16	-0.30	0.25	-0.22			
Kaviar	-0.36	-0.31	0.19	-0.27	-0.13	-0.26	0.21	-0.18	0.77		
Eigen	0.85	0.89	-0.63	0.89	0.78	0.81	-0.59	0.61	-0.27	-0.22	

Tabla 4. Correlación de Pearson entre las columnas del set de datos pre-filtrado (A) y post-filtrado (B). En rojo (🔴) se muestran las variantes altamente correlacionadas (>= 0.80).

Entrenamientos:

Por un lado, se presentan las métricas de error MAE y RMSE para las regresiones lineales y por otro, el porcentaje de error con el que se clasificó cada clase en los modelos de AA. Procedemos a comparar horizontalmente los rendimientos de las diferentes particiones sobre el set de validación interno a los efectos de definir el escenario con mejor desempeño en identificar variantes patogénicas.

Escenario Significancia incierta - Significancia cierta

Según la validación sobre el set de entrenamiento, un modelo de Regresión Lineal entrenado con el escenario Significancia incierta - Significancia cierta genera predicciones con MAE de 0.45 y RMSE de 0.47 (Tabla 6. B). En cuanto a los modelos de AA, ninguno parece generar predicciones confiables con el escenario SI-SC. Vemos que los modelos basados en Arboles generan errores entre 20% y 30% para ambas clases (Tabla 5. A, C, D - b). Mientras que los modelos basados en Maquinas de Soporte de Vectores y Redes Neuronales generan predicciones con errores aun mayores, superiores a 30% en ambas clases para los tres modelos (Tabla 5. E, F, G - b).

A

Métrica	MAE	RMSE
Error	0.19	0.24

B

Métrica	MAE	RMSE
Error	0.45	0.47

C

Métrica	MAE	RMSE
Error	0.18	0.24

Tabla 6. Métricas de error al validar el modelo de Regresión Lineal entrenado en el escenario Todas las clases (A), Significancia incierta - Significancia cierta (B) y Benignas - Patogénicas (C). El error se normalizó por el rango de la variable resultado.

A

Clase	B	LB	US-B	US	US-P	LP	P
Error	42.71	33.88	97.85	75.78	99.28	53.48	37.64

B

Clase	SI	SC
Error	28.72	29.50

C

Clase	B	P
Error	3.30	2.07

Tabla 7. Promedio de error (%) por clase para los modelos generados en el escenario de todas las clases (A), Significancia incierta - Significancia cierta (B), y Benignas - Patogénicas (C).

Escenario Todas las clases

En cuanto al escenario TC, el modelo de Regresión Lineal resultante genera predicciones con un error considerablemente menor al escenario anterior, un MAE de 0.19 y RMSE de 0.24 (Tabla 6. A). Las matrices de confusión sin embargo muestran errores muy altos para todos los modelos de AA entrenados con el escenario TC. Los modelos de Árbol generan predicciones decentes para las variantes Benignas, con un error cercano al 25%, pero predicciones muy malas en variantes patogénicas, con un error próximo al 50% en los tres clasificadores (Tabla 5. A, C, D - a). Este es también el caso en el resto de los clasificadores, con errores entre el 28% y 78% para ambas clases (Tabla 5, B, E, F, G - a). En el caso particular de SVM Polynomial vemos una muy buena predicción de variantes patogénicas, con un error del 4%, pero una predicción mala en variantes benignas, con un error del 78%, lo cual nos indica que el clasificador esta simplemente prediciendo todas las variantes como patogénicas.

Escenario Benigna-Patogénicas

Por último, el modelo de Regresión Lineal entrenado con el escenario B-P presenta MAE y RMSE similares a los del escenario anterior, 0.18 y 0.24 respectivamente (Tabla 6. C). Sin embargo, las matrices de confusión muestran que este escenario genera modelos ampliamente superiores en la identificación de variantes patogénicas. El error es particularmente bajo en la predicción generada con los modelos basados en Árbol y el modelo de SVM Polynomial. En los cuatro casos no supera el 2% de error para ambas clases (Tabla 5. A, C, D, E - c). El modelo de Vecinos mas Cercanos y el modelo SVM Radial generaron buenas predicciones cuando la comparamos a las generada con las otras particiones, estas se encuentran cercanas al 3% en ambas clases para ambos modelos (Tabla 5. B, F - c). Por último, el modelo de Redes Neuronales mejoró su predicción cuando la comparamos con el modelo equivalente entrenado con las particiones SI-SC y TC. Sin embargo, es el modelo de peor desempeño dentro del escenario B-P, con errores del 15.1% y 3.4% para variantes Benignas y Patogénicas respectivamente (Tabla 5. G - c).

El error promedio de los 7 modelos generados a partir del escenario Benignas - Patogénicas para clasificar variantes Patogénicas es de 2.07%, y para clasificar variantes Benignas es 3.30% (Tabla 7. C). Notoriamente

inferior al producido por los modelos entrenados con las otras dos particiones, donde ninguna de las clases se predice con un error inferior al 28% (Tabla 7. A y B).

Al estructurar el set de entrenamiento de manera binaria y sin incluir las variantes de Significado incierto, la precisión del clasificador aumenta notoriamente. Ésta genera mejores resultados de validación interna para todos los algoritmos. Algunos de ellos sin embargo, son mejores que otros dentro de este escenario.

El más preciso en validación interna, es sin duda el modelo generado por Gradient Boosted Tree (Tabla 5. D, c), éste obtuvo 0% de error para ambas clases (Figura 5). Éste clasificó correctamente las 4.000 variantes Benignas y las 4.000 variantes Patogénicas. Sin embargo, como mencionamos anteriormente, este es su desempeño sobre el set de entrenamiento y una predicción perfecta es señal de sobre-ajuste. Por ello, y sabiendo que los Árboles de decisión son particularmente propensos al sobre-entrenamiento (excepto Bosques Aleatorios), sería apresurado afirmar que el modelo de GBT es el que mejor interpreta el escenario de datos Benignas - Patogénicas.

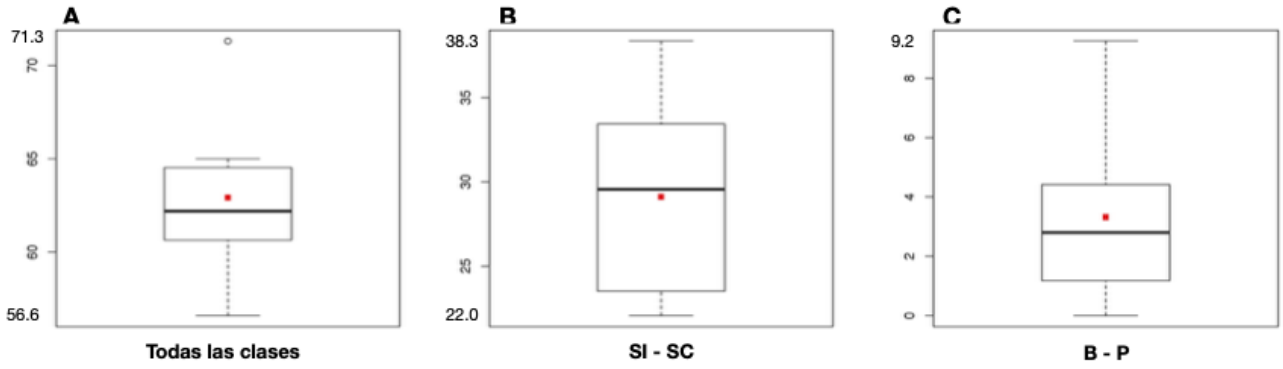


Figura 5. Diagramas de caja mostrando la distribución del error general en los modelos generados en el escenario de todas las clases (A), Significancia incierta - Significancia cierta (B), y Benignas - Patogénicas (C).

A

Métrica	MAE	RMSE
Error	0.19	0.24

B

Métrica	MAE	RMSE
Error	0.46	0.48

C

Métrica	MAE	RMSE
Error	0.19	0.26

Tabla 8. Métricas de error al evaluar el modelo de Regresión Lineal entrenado en el escenario Todas las clases (A), Significancia incierta - Significancia cierta (B) y Benignas - Patogénicas (C). El error se normalizó por el rango de la variable resultado.

a

A. Árboles de decisión

	B	LB	US-B	US	US-P	LP	P	Error
B	1587	443	39	186	1	26	21	20.6
LB	340	1345	106	794	2	9	17	32.7
US-B	1	1	16	4	0	3	1	92
US	49	188	32	604	8	63	62	69.8
US-P	0	0	0	0	0	0	0	100
LP	11	12	4	204	5	1207	734	39.6
P	12	11	3	208	4	692	1165	41.7

b

	SI	SC	Error
SI	1731	336	13.4
SC	769	2164	30.7

c

	B	P	Error
B	3960	37	0.9
P	40	3963	1

a

B. Vecinos más cercanos

	B	LB	US-B	US	US-P	LP	P	Error
B	1079	346	41	258	0	14	16	46.2
LB	599	1261	91	562	1	29	22	36.9
US-B	0	0	1	1	0	0	0	99.5
US	243	308	50	611	4	67	82	69.4
US-P	0	0	0	0	0	0	0	100
LP	40	41	8	293	6	1181	730	40.9
P	39	44	9	275	9	709	1150	42.5

b

	SI	SC	Error
SI	1778	760	30.4
SC	722	1740	28.8

c

	B	P	Error
B	3860	120	3
P	140	3880	3.5

a

C. Bosques aleatorios

	B	LB	US-B	US	US-P	LP	P	Error
B	1468	384	1	121	0	18	8	26.6
LB	461	1022	7	485	0	3	4	48.9
US-B	36	85	13	59	0	3	4	93.5
US	179	650	5	695	1	276	194	65.2
US-P	0	1	0	6	0	9	4	100
LP	12	6	0	56	0	1151	775	42.4
P	5	9	0	62	0	908	1016	49.2

b

	SI	SC	Error
SI	1841	659	26.3
SC	445	2055	17.8

c

	B	P	Error
B	3936	64	1.6
P	49	3951	1.2

a

D. Gradient Boosted Tree

	B	LB	US-B	US	US-P	LP	P	Error
B	1466	415	26	162	0	18	9	26.7
LB	433	1255	111	805	1	8	7	37.2
US-B	0	0	0	0	0	0	0	100
US	54	280	50	497	5	59	80	75.1
US-P	0	0	0	0	1	0	0	95
LP	40	36	9	274	10	1142	846	42.9
P	7	14	4	262	3	773	1058	47.1

b

	SI	SC	Error
SI	1844	593	23.7
SC	656	1907	26.2

c

	B	P	Error
B	4000	0	0
P	0	4000	0

a

E. SVM Polynomial

	B	LB	US-B	US	US-P	LP	P	Error
B	424	22	1	2	0	0	0	78.8
LB	1189	1425	114	895	1	27	20	28.7
US-B	0	0	0	0	0	0	0	100
US	145	191	19	181	1	16	19	90.9
US-P	0	0	0	0	0	0	0	100
LP	17	34	9	46	1	57	59	97.1
P	225	328	57	876	17	1900	1902	4

b

	SI	SC	Error
SI	1664	1084	43.3
SC	836	1416	33.4

c

	B	P	Error
B	3938	34	0.8
P	62	3966	1.5

a

F. SVM Radial

	B	LB	US-B	US	US-P	LP	P	Error
B	898	131	7	50	0	2	2	55.1
LB	922	1567	134	933	2	11	14	21.6
US-B	0	0	0	0	0	0	0	100
US	113	231	38	417	5	57	64	79.1
US-P	0	0	0	0	0	0	0	100
LP	47	49	17	364	9	1158	932	42.1
P	20	22	4	236	4	772	988	50.6

b

	SI	SC	Error
SI	1796	775	31
SC	704	1725	28.1

c

	B	P	Error
B	3844	71	1.7
P	156	3929	3.9

a **G. Redes neuronales**

	B	LB	US-B	US	US-P	LP	P	Error
B	1099	687	0	138	0	20	56	45
LB	257	1376	0	290	0	19	55	31.2
US-B	15	119	0	42	0	7	17	100
US	92	891	0	379	0	181	457	81
US-P	0	0	0	5	0	5	10	100
LP	0	11	0	72	0	611	1306	69.4
P	3	12	0	67	0	487	1431	28.4

b

	SI	SC	Error
SI	1839	661	33
SC	830	1670	41.5

c

	B	P	Error
B	3698	302	15.1
P	68	3932	3.4

Tabla 5. Matrices de confusión correspondientes a la validación de cada modelo entrenado con los escenarios de Todas las clases (a), Significancia incierta - Significancia cierta (b) y Benignas - Patogénicas (c). Las abreviaciones en a y c corresponden a Benignas (B), Posiblemente benignas (LB), Significancia incierta / Benignas (US-B), Significancia incierta (US), Significancia incierta / Patogénica (US-P), Posiblemente patogénica (LP), Patogénica (P). Las abreviaciones en b corresponden a Significancia incierta (SI) y Significancia cierta (SC). Los valores de error se muestran en porcentaje (%).

Otros tres clasificadores obtuvieron rendimientos muy buenos sobre el set de validación, éstos son los Árboles de Decisión (Tabla 5. A, c), el de Bosques Aleatorios (Tabla 5. C, c) y el de SVM Polynomial (Tabla 5. E, c). Interesantemente, otros dos clasificadores basados en árboles modelan de manera adecuada nuestro set de datos. Por un lado, los Árboles de Decisión tuvieron un error general muy bajo de 0.95% (Tabla 5. A, c). Sin embargo, como los GBT, estos son propensos al sobre-ajuste por lo que hay que estudiar también su error con el set de evaluación para poder determinar con certeza que tan bien interpreta los datos este modelo. Por otro lado, los Bosques Aleatorios buscan evitar el sobre-ajuste que caracteriza a los Árboles de Decisión. Por esta propiedad y dado que presenta un error general de 1.4% (Tabla 5. C, c), es un modelo muy interesante a seguir estudiando y ver como se comporta con el set de evaluación. Por último, el SVM Polynomial es el único modelo no basado en árboles que presentó muy buena precisión sobre el set de validación. Con un error general del 1.15% (Tabla 5. E, c) es el tercer modelo más preciso.

Los cuatro mejores modelos son entonces GBT, Árboles de Decisión, SVM Polynomial y Bosques Aleatorios en ese orden para el escenario B-P. Interesantemente, cuando comparamos estos cuatro modelos para las demás particiones no se destacan del resto (Tabla 5. A y C-E, a y b).

Según los resultados de validación, nuestro clasificador alcanza su desempeño óptimo al entrenarlo con un set binario sin variables de Significado incierto. Además, vimos que tres de los cuatro mejores modelos se basan en Árboles de Decisión. Para tomar en cuenta el sobre-ajuste y evaluar el desempeño del clasificador sobre variantes nuevas, debemos considerar las predicciones de cada modelo sobre el set de evaluación.

Evaluación del escenario B-P:

El modelo de Regresión Lineal evaluado con el set de datos nuevo genera predicciones similares a las observadas dentro del set de entrenamiento. Se puede observar un RMSE considerablemente superior al MAE (Tabla 8). Pasando a la evaluación de los modelos de AA, vemos que los clasificadores GBT, Árboles de decisión, SVM Polynomial y Bosques aleatorios generan las predicciones mas precisas. Todos ellos presentan errores menores al 1.6% en la clasificación de variantes Patogénicas y errores menores al 2.7% en la clasificación de variantes benignas (Tabla 9. A, C, D, E). Los modelos GBT y RF se destacan del resto, con

errores promedio de 1.10% y 1.25% respectivamente, sensibilidades de 0.99 y 0.98 respectivamente y ambos presentan un área debajo de la curva ROC de 0.99 (Figura 7. A y C).

A

	B	P	Error
B	1062	30	2.7
P	3	418	0.7
Sensibilidad			0.97
Especificidad			0.99

B

	B	P	Error
B	1053	39	3.5
P	10	411	2.3
Sensibilidad			0.96
Especificidad			0.97

C

	B	P	Error
B	1079	13	1.1
P	6	415	1.4
Sensibilidad			0.98
Especificidad			0.98

D

	B	P	Error
B	1083	9	0.8
P	6	415	1.4
Sensibilidad			0.99
Especificidad			0.98

E

	B	P	Error
B	1068	24	2.1
P	7	414	1.6
Sensibilidad			0.97
Especificidad			0.98

F

	B	P	Error
B	1061	31	2.8
P	6	415	1.4
Sensibilidad			0.97
Especificidad			0.98

G

	B	P	Error
B	1023	69	6.3
P	4	417	0.9
Sensibilidad			0.99
Especificidad			0.93

Tabla 9. Matrices de confusión correspondientes a la evaluación de los modelos Arboles de decisión (A), Vecinos mas cercanos (B), Bosques aleatorios (C), Gradient Boosted Trees (D), SVM Polinominal (E), SVM Radial (F), Redes neuronales (G). Los valores de error se muestran en porcentaje (%).

Para estudiar el sobreajuste de los cuatro modelos con alto rendimiento (Arboles de Decisión, GBT, RF y SVM Polynomial) vamos a comparar los resultados obtenidos en evaluación (Tabla 9) con aquellos obtenidos durante la validación (Tabla 5 y 8). Por un lado, el modelo GBT obtuvo excelentes resultados (0% error) en validación (Tabla 5. D, c), pero sabemos que es propenso al sobre-ajuste. Como era de esperar, su error general aumenta al predecir variantes nuevas. En este caso, el error general pasa de 0% a 1.1% (0.8% en variantes Benignas y 1.4% en variantes Patogénicas) (Tabla 9. D, c). Es probable que este aumento de 1.1 puntos porcentuales de error (Figura 6) se deba a que el clasificador este sobre-ajustado a nuestro set de entrenamiento. Por otro lado, el modelo de Bosques Aleatorios generó una buena predicción en validación con un error general de 1.4% (Tabla 5. C, c), mientras que con el set de evaluación obtuvo un error general de 1.25% (Tabla 9. C, c) (1.1% en variantes Benignas y 1.4% en variantes Patogénicas). Estos buenos resultados en evaluación son esperables ya que dichos modelos reducen el sobreajuste de los arboles de decisión dejando columnas afuera en la formación de cada árbol. Estos resultados indican que el clasificador puede predecir variantes nuevas (no etiquetadas) con la misma precisión que lo hace sobre variantes conocidas (sets de validación y evaluación). El tercer modelo más preciso sobre datos nuevos son los Árboles de Decisión. Éstos obtuvieron un error muy bajo para el set de evaluación (0.95%), pero sabemos que son modelos propensos al sobre-ajuste. Esto se ve reflejado en el error general de 1.7% (2.7% en variantes Benignas y 0.7% en variantes Patogénicas) que obtuvieron para el set de evaluación (Tabla 9. A, c), 0.75 puntos porcentuales mayor al anterior (Figura 6). Por ultimo, el error promedio en validación del modelo SVM Polynomial fue de 1.15% (Tabla 5. E, c), mientras que con el set de evaluación su error promedio pasó a ser 1.85% (2.1% en variantes Benignas y 1.6% en variantes Patogénicas) (Tabla 9. E, c), con un aumento de 0.7 puntos porcentuales (Figura 6). La precisión del modelo, pero lo hace de manera menos drástica que los modelos de Árboles de decisión y GBT. Parecería entones ser que el modelo no sufre de sobre-ajuste considerable.

De el estudio de sobre-ajuste se destacan dos clasificadores, GBT y RF. Por un lado, el modelo de Gradiente Boosted Tree generó una predicción perfecta sobre el set de validación y la mejor de las predicciones sobre el set de evaluación. Si bien parece estar ligeramente sesgado por sobre-ajuste (Figura 6), generó una predicción muy buena sobre datos nuevos. Por otro lado, el modelo de Bosques Aleatorios generó una predicción decente

sobre ambos sets de datos (validación y evaluación). Por estas razones, tanto el modelo de GBT como el de RF tienen la capacidad de generar predicciones sobre variantes no-etiquetadas de un nuevo juego de datos con precisión similar a la vista sobre el set de evaluación.

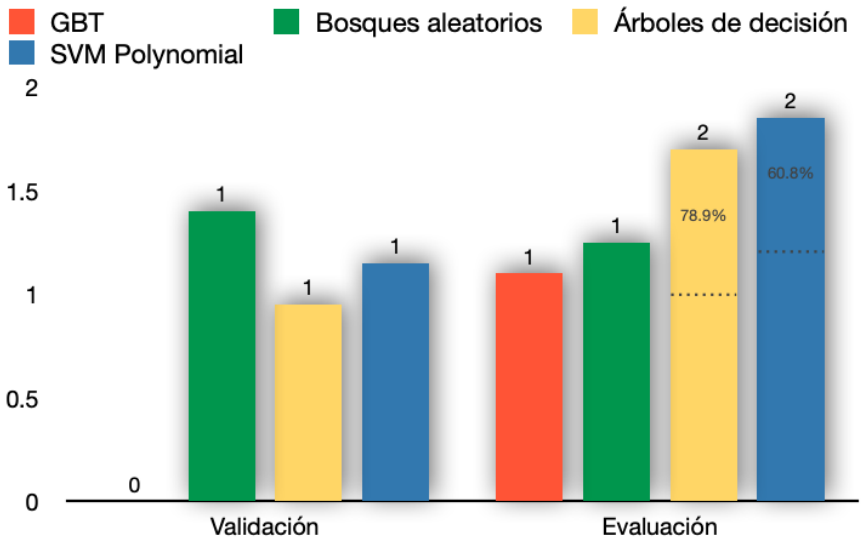


Figura 6. Histogramas mostrando el error general de los cuatro algoritmos más precisos para el escenario B-P según el set de validación (izquierda) y el set de evaluación (derecha). El error general se calcula como el promedio de los errores por clase.

Por el contrario, el estudio del sobre-ajuste sobre los modelos de Árboles de Decisión y SVM Polynomial indica que su precisión al predecir variantes de pacientes uruguayos puede divergir considerablemente de lo observado sobre el set de evaluación. Ambos obtuvieron un error general cerca de 50% mayor al obtenido con GBT para el set de evaluación. Y un aumento por sobre-ajuste de 0.75 y 0.7 puntos porcentuales respectivamente (Figura 6), lo que representa 78.9% y 60.8% más error en cada caso respectivamente. Por estas dos razones los análisis posteriores se realizarán únicamente sobre los modelos GBT y RF.

Los dos modelos de mejor rendimiento (Bosques aleatorios y GBT) son los únicos que predicen mejor variantes Benignas que variantes Patogénicas (Tabla 9. C-D, c). Los demás modelos clasifican erróneamente mas variantes Benignas que variantes Patogénicas (Tabla 9. A-B y E-G, c). Las posibles causas de esto se discuten a continuación (ver Discusión).

Habiendo identificado dos modelos con buen rendimiento, pasamos a evaluar sus predicciones sobre diferentes tipos de variantes. Para ello se produjeron 4 sets de variantes diferentes, uno con variantes Benignas y Patogénicas (BP), otro con variantes Posiblemente benignas y Posiblemente patogénicas (LBLP), otro con variantes únicamente de Significado incierto (US) y el último con variantes de Significado incierto / Benignas y Significado incierto / Patogénicas (USBP) (ver Métodos, figura 4. A).

En el set de evaluación LBLP, el área debajo de la curva ROC disminuye para ambos modelos (Figura 7). En el caso de los Bosques Aleatorios, el AUROC pasa de 0.98 a 0.96 (Figura 7. A-B), con un descenso de 0.0197. Mientras que en el Gradient Boosted Tree el AUROC pasa de 0.98 a 0.97 (Figura 7. C-D), con un de 0.0186, ligeramente menor al de RF. Además, la predicción en variantes Posiblemente benignas y Posiblemente patogénicas parece afectar solo su capacidad de predecir variantes Patogénicas. La Sensibilidad disminuye 0.03 puntos en ambos casos, mientras que la Especificidad se mantiene constante en el modelo de RF y aumenta 0.01 puntos en el modelo de GBT (Tabla 9. C-D, c y Tabla 10. A-B, a). Los histogramas de la figura 8. B nos muestran que el modelo GBT predice ambas clases con mayor confianza que el modelo RF. En este último se observa una distribución de probabilidad con colas para ambas clases. Esto indica que ciertas variantes se predicen correctamente, pero con menor confianza que el modelo GBT.

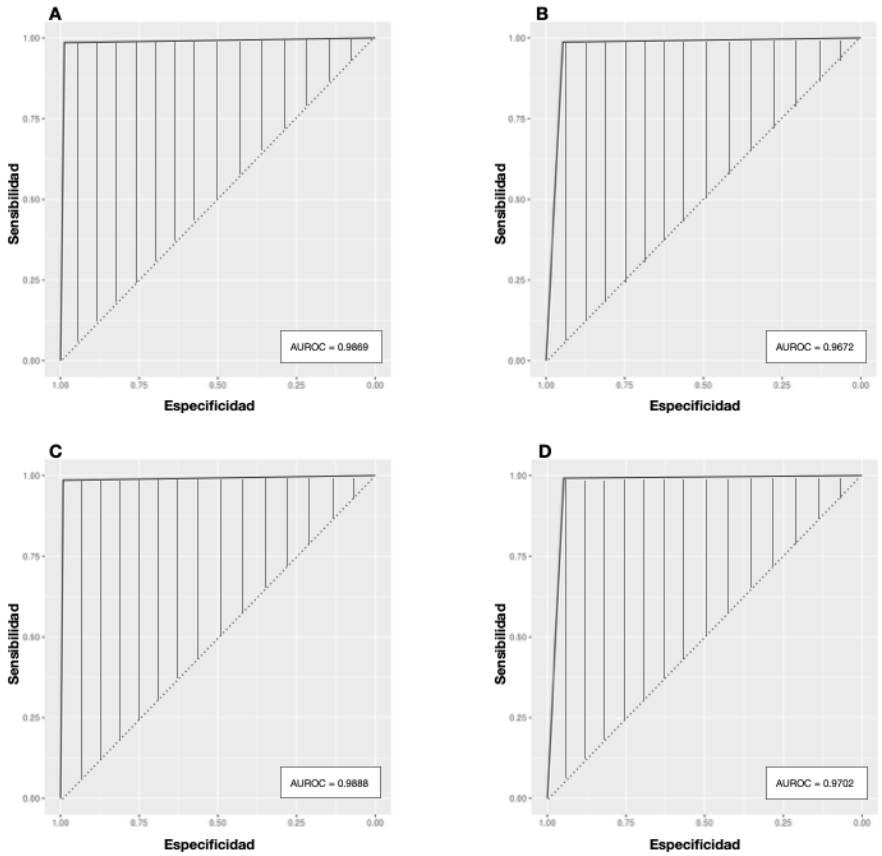


Figura 7. Curvas ROC al predecir variantes Benignas - Patogénicas con los modelos RF (A) y GBT (C), y variantes Posiblemente benignas - Posiblemente patogénicas con los modelos RF (B) y GBT (D).

Es importante poder clasificar variantes etiquetadas con Significancia incierta según ClinVar. Al no estar anotadas, no se pueden computar métricas de error para esta clasificación, pero si se puede evaluar la confianza con la que cada modelo las separa. Sobre esta línea, el modelo de GBT vuelve a superar al modelo de RF. En los histogramas vemos que las predicciones generadas por el modelo GBT se distribuyen en dos picos bien marcados, uno en el 0% de confianza de ser patogénica (es decir variantes Benignas) o otro en el 100% de confianza de ser patogénica (es decir variantes Patogénicas) (Figura 8. C). Se pueden observar apenas unas pequeñas colas (n < 100) que corresponderían a variantes cuya clasificación no es confiable. El modelo entrenado con Bosques Aleatorios presenta una distribución totalmente diferente. Si bien se observan dos picos claros (en 0% y 100%) que indican cierta separación de los datos, gran parte de las variantes (~ 52.9%) se encuentran en un espectro de confianza que oscila entre el 10% y el 90% (Figura 8. C). Por ello, no podemos confiar en la predicción que este modelo genera sobre el 50% de las variantes de Significado incierto.

A. Gradient Boosted Tree

a

	LB	LP	Error
LB	1429	32	2.1
LP	78	4043	1.7
Sensibilidad			0.94
Especificidad			0.99

b

	US-B	US-P	Error
US-B	251	3	1.1
US-P	27	19	58.6
Sensibilidad			0.90
Especificidad			0.86

B. Bosques aleatorios

a

	LB	LP	Error
LB	1427	51	3.3
LP	80	4024	1.8
Sensibilidad			0.94
Especificidad			0.98

b

	US-B	US-P	Error
US-B	248	6	2.3
US-P	26	20	56.5
Sensibilidad			0.89
Especificidad			0.90

Tabla 10. Matrices de confusión obtenidas al evaluar los modelos GBT (A) y RF (B) con variantes Posiblemente benignas - Posiblemente patogénicas (a) y. Significado incierto / Benignas - Significado incierto / Patogénicas (b).

El set con variantes de Significado incierto / Benignas y Significado incierto / Patogénicas (USBP), nos permite determinar los SNPs de Significancia incierta que fueron mal clasificados y así confirmar lo visto en la

separación de datos (Tabla 10. A-B, b). En este análisis, ambos modelos mostraron buen desempeño en clasificar variantes de Significado incierto / Benignas. Para GBT, la clasificación fue incluso mejor que al evaluar con variantes Posiblemente benignas, confirmando los resultados obtenidos al evaluar la separación de variantes de Significado incierto y la confianza con la que este modelo clasifica variantes genómicas. El modelo RF también fue mas eficiente al clasificar variantes US-B que variantes LB (Tabla 8. B), pero distribución de la confianza de clasificación muestra una cola bastante grande con variantes de Significado incierto / Benignas a las que se le asignan probabilidades de patogenicidad cercanas al 90% (Figura 8. D). Por otro lado, vemos que ambos modelos clasificaron de manera adversa las variantes Significado incierto / Patogénicas (Tabla 10. A y B - b). El mal resultado obtenido puede deberse al número reducido de variantes US-P disponibles para evaluación (46) o a un problema en la anotación de ClinVar.

Priorización de variantes en genomas uruguayos

Se consideró un paciente del proyecto URUGENOMES que no pude ser diagnosticado a través de su exoma. Se obtuvieron 10.000.000 de variantes correspondientes a todos los pacientes del proyecto URUGENOMES. De estas se mantuvieron únicamente 4.683.462 variantes no homocigotas recesivas correspondientes al paciente ER23. De estas, 24.077 variantes codificantes fueron filtradas, por lo que se obtuvieron 4.659.385 variantes no codificantes a anotar. De estas, 47.249 variantes fueron anotadas con mas de 40% de las columnas requeridas, siguiendo los mismos parámetros por los que se filtraron las variantes en ClinVar. De las variantes restantes, el modelo RF predijo únicamente 6 variantes no-codificantes como patogénicas con una confianza mayor o igual al 95% (Tabla 11).

Cuatro de estas mutaciones se encuentran en regiones intergénicas, de éstas, tres están asociadas al gen FRG2C y una al gen FRG1. De las dos restantes, una se encuentra en la región intrónica del gen SRSF10 y la otra en una región transcripta no codificante (ncARN) del gen TEKT4P2. Cinco de estas mutaciones consisten en una transición de Guanina a Adenina, mientras que una de ellas es una transversión de una Guanina a una Citosina. Las más interesantes son las dos relacionadas con el gen SRSF10 y TEKT4P2.

Discusión

Imputación:

Cuando analizamos los resultados de imputación vemos que claramente el algoritmo basado en KNN produce mejores predicciones en la mayoría de las columnas (Tabla 3).

En el caso de fitCons y gnomAD, dicho algoritmo genero la mejor predicción según el Error Absoluto Medio (MAE) pero no así tomando en cuenta la Raíz Cuadrada del Error Cuadrático Medio (RMSE). Es posible que dicha medida tome en cuenta valores extremos que el MAE no es capaz de capturar. Esto indica que KNN produce una buena predicción para la mayoría de las variantes (capturadas por el buen resultado en MAE) pero errores importantes en un numero reducido de variantes (capturados por RMSE).

Por otro lado, gnomAD y Kaviar, ambas columnas de frecuencia alélica, obtuvieron una buena predicción por parte de KNN. RMSE capturó errores importantes en ciertas variantes imputadas de gnomAD, esto puede deberse a variantes de alta/baja frecuencia poblacional que hayan sido imputadas con una frecuencia cercana a 0.5 lo cual produciría un error considerable en cualquiera de los casos. Por otro lado, Kaviar tuvo una mala imputación al tomar en cuenta tanto MAE como RMSE, superada en ambos casos por el algoritmo *mi*. Interesantemente, vemos que el error capturado por RMSE para esta columna es mucho mayor al capturado por MAE, siguiendo la línea de los mencionado anteriormente. Por la naturaleza del KNN que utiliza vecinos más cercanos, puede ser que ciertas variantes de frecuencia alta/baja sean imputadas con un valor cercano al promedio según el valor de vecinos que se tome para la predicción.

En líneas generales el acercamiento utilizado para identificar el algoritmo más optimo con el cual imputar nuestros datos no solo arrojo resultados concluyentes, sino que, como era de esperar dada la naturaleza de los datos, supero ampliamente el protocolo tradicional de imputar por el promedio.

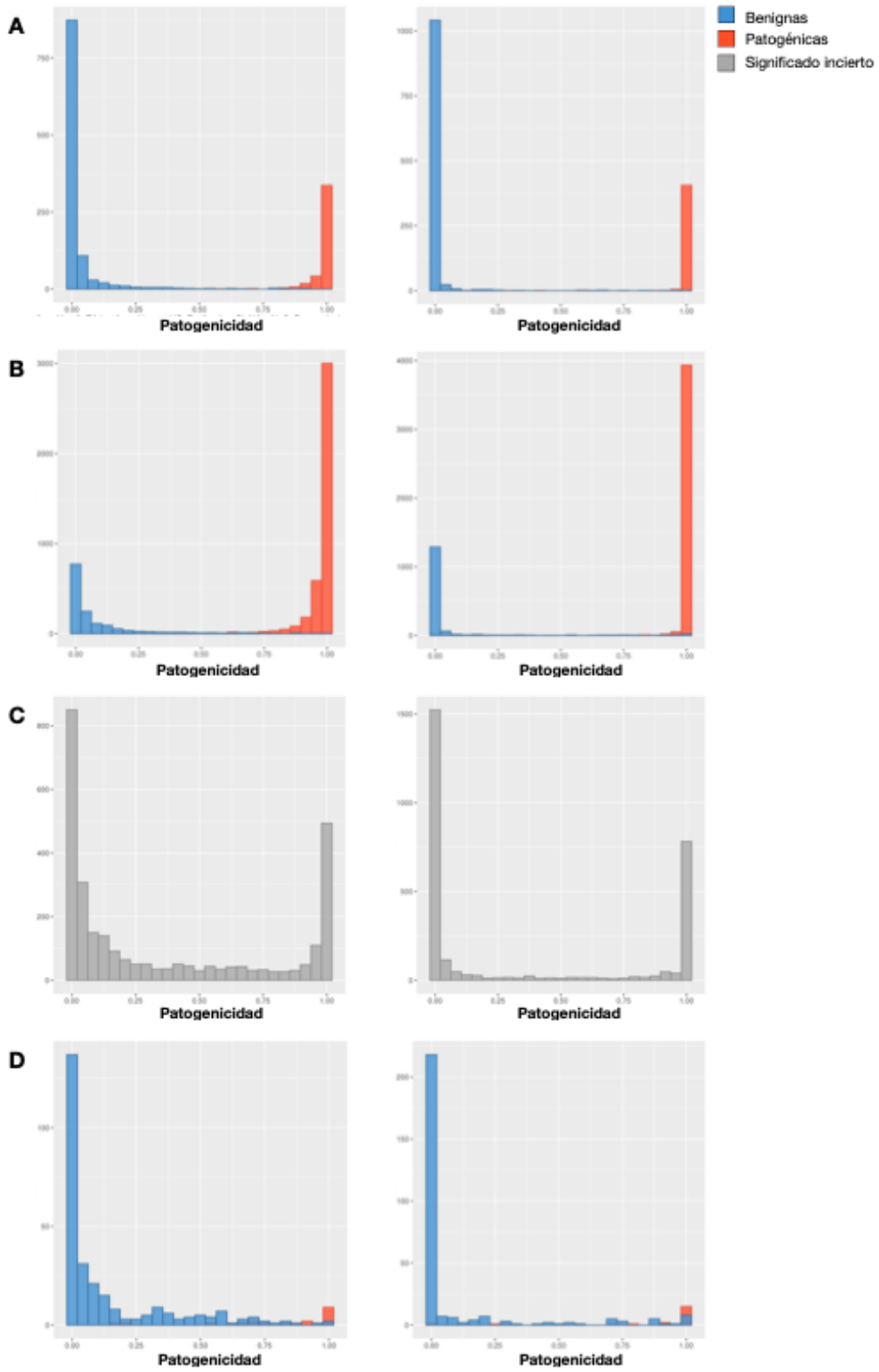


Figura 8. Histogramas de la confianza con la que los modelos RF (izquierda) y GBT (derecha) clasifican como Patogénica una variante. Las predicciones se realizaron sobre los sets de evaluación BP, LBLP, US y USBP (A, B, C y D respectivamente).

Entrenamiento:

Los resultados obtenidos para los escenarios, tanto Significancia incierta - Significancia cierta como Todas las clases fueron deficientes sin importar el algoritmo utilizado. Esto puede deberse a la característica biológicas que diferencian una variante Benigna de una Patogénica (frecuencia poblacional, conservación evolutiva, entre otras). Cualquier algoritmo de Aprendizaje Automático se basa en dichas características para separar los datos, ya sea entre SI-SC, cada clase (TC) o B-P. Cuando asignamos las variantes Patogénicas y Posiblemente patogénicas a diferentes categorías le estamos separando dos variantes con características biológicas muy similares. Y viceversa, cuando ponemos variantes Benignas y Patogénicas en una misma categoría (Significancia cierta) entrenamos algoritmo para que agrupe dos variantes con características biológicas opuestas. Por esta razón, el escenario que mejor funciona es aquel que busca separar dos variantes biológicamente diferentes en dos categorías independientes (B-P). Esto se puede ver claramente en la Tabla 7 donde tanto para el escenario SI-SC cómo para TC el error promedio de los 7 algoritmos es considerablemente mayor al producido por el escenario B-P. Podemos incluso ver que los errores producidos al asignarle diferente categoría a variantes con características biológicas similares (TC) produce una peor predicción (Tabla 7. A).

Evaluación:

En línea con la bibliografía publicada recientemente[35-37], los dos algoritmos óptimos para interpretan patogenicidad en variantes de nucleótido simple en regiones no codificantes son los RF y los GBT. Es de esperar que una red neuronal consiga un rendimiento similar, en el presente proyecto no se realizó el ajuste de parámetros y el entrenamiento necesario para ello. Pero cabe remarcar que en una extensión del mismo se debería hacer mayor hincapié en el entrenamiento de redes neuronales mas allá del MLP, como pueden ser las LSTM.

En las clasificaciones binarias se toma una de las dos clases como objeto a predecir o clase positiva, en nuestro caso esta sería la variante Patogénica ya que es preferible predecir muchas variantes como Patogénicas y luego filtrar manualmente los falsos positivos, que el contrario. La tabla 9 nos muestra que tanto RF como GBT consiguen la mejor predicción de variantes patogénicas en comparación al resto de los modelos.

Si bien ambos se basan en Arboles de Decisión, generan su predicción final usando procedimientos diferentes. Los algoritmos basados en RF entrenan varios Arboles de Decisión a partir de un numero reducido de variables (mtry), la predicción final se computa votando la predicción de todos los árboles. Al no utilizar todos los predictores (columnas) que componen el set de datos, estos algoritmos reducen considerablemente el sobreajuste. Al ser entrenado cada árbol sobre un set de datos diferente el modelo no estaría sobre ajustado a un único set de datos, en consecuencia, es de esperar que el error de estos algoritmos no aumente considerablemente al clasificar datos nuevos. Por otro lado, los algoritmos basados en GBT toman los residuales del ultimo árbol entrenado como “input” para el nuevo árbol e iteran el proceso “n” veces. El sobreajuste de estos algoritmos se puede regular con el llamado “radio de aprendizaje” por el cual se multiplican los residuales en cada iteración. Sin embargo, los modelos GBT suelen generar buenas predicciones sobre el set de entrenamiento, pero no así sobre un set de datos nuevo, por lo que la precisión de su predicción puede variar al clasificar variantes nuevas. En nuestro caso particular esta es una diferencia importante, y pone a RF en ventaja por sobre GBT, ya que el error que pueda generar sobre los pacientes (por ejemplo de URUGENOMES) seria mas cercano a su error sobre el set de evaluación, en comparación a GBT. En este último, su error sobre el set de validación no es indicador confiable de la precisión con la que pueda clasificar variantes de nuevos pacientes.

Por otro lado, la tabla 9 nos muestra que el modelo basado en RF es el que consigue los segundos mejores resultados sobre el set de evaluación, solo ligeramente por detrás de GBT. Tendiendo en cuenta lo discutido anteriormente, RF sería el modelo más adecuado para predecir variantes patogénicas en el genoma de los pacientes de URUGENOMES. Si bien GBT muestra mejores resultados sobre el set de evaluación, no podemos estar seguros que su desempeño vaya a ser similar sobre dichos pacientes.

Vemos además que los modelos de árbol sin muestreo aleatorio en los predictores (Arboles de Decisión y GBT) aumentan considerablemente su error al pasar del set de entrenamiento (Tabla 5. A y D - c) al set de evaluación (Tabla 9. A y D - c), incluso por encima de SVM (Figura 6). Esto concuerda con lo esperado y lo discutido previamente, apoyando el uso de RF para predecir variantes patogénicas en nuestros pacientes.

El escenario Benignas-Patogénicas genera las mejores predicciones, pero también puede ser útil para clasificar variantes de Significado incierto. Estas son particularmente interesantes para nuestro proyecto ya que la amplia mayoría de los SNPs en regiones no-codificantes de nuestros pacientes van a estar etiquetadas como Significado incierto según ClinVar. Al generar predicciones binarias, este escenario restringe su clasificación a Benigna o Patogénica. Si bien no es posible evaluar la precisión de su clasificación en variantes inciertas, se puede usar la confianza de cada predicción como indicador del desempeño del modelo en separar dichas mutaciones. De esta manera podemos filtrar aquellas variantes que hayan sido predichas cómo Patogénicas con una confianza menor a cierto umbral (ej. 95%), priorizando así un numero manejable de variantes, posiblemente inferior a 100.

El análisis de variantes US-B / US-P nos permitió ver que el modelo es capaz de clasificar correctamente variantes inciertas (Tabla 10. A y B - b, Figura 8. D). Sin embargo, el tamaño muestral no es suficiente para obtener resultados concluyentes.

El bajo numero de variantes de Significado Incierto - Patogénicas (46) no permite determinar la sensibilidad del modelo hacia esta clase. Además, al contar con un numero reducido de entradas, una falla en la anotación de ClinVar puede afectar considerablemente las métricas de error calculadas para esta clase. Éste no es el caso con las variantes de Significado Incierto - Benignas ya que el número de entradas (493) permite calcular la sensibilidad de ambos modelos (Tabla 10. A y B - b).

Variantes priorizadas en genomas uruguayos

Al momento de estudiar en detalle cada variante se deben de tener en cuenta su posición genómica, proceso biológico en el que participa según el fenotipo observado en el paciente y referencias bibliográficas donde se haya estudiado los efectos producidos por dicha mutación.

El gen SRSF10 codifica para una proteína perteneciente a la familia de proteínas serina-arginina que participan y regulan el splicing del ARN. Se ha reportado un un ciclo celular alterado en células deficientes de dicho gen, llegando a la conclusión que el mismo cumple un rol fundamental en la respuesta a estrés inhibiendo la maquinaria de splicing[110]. La variante intrónica G>C bien podría afectar el mecanismo de transesterificación impidiendo la traducción de la proteína.

El gen TEKT4P2 codifica para una proteína que interactúa con MAFF. El proyecto Deciphering Developmental Disorders (DDD) del Reino Unido tiene el objetivo de identificar las bases génicas de enfermedades del desarrollo[111]. Hasta la fecha dicho estudio ha identificado 13 pacientes cuya patología puede explicarse por mutaciones en el gen TEKT4P2. Si bien ninguna de éstas corresponde a la variante priorizada por nuestro clasificador, esta claro que un error en la transcripción o traducción de dicho gen puede conducir a una enfermedad congénita.

Tres de las seis variantes priorizadas se encuentran en regiones intergénicas asociadas al gen FRG2C. Se ha visto que expresión diferencial en ARN largo no-codificante de dicho gen puede estar asociada con diferenciación osteogénica anormal en el desarrollo de espondilitis anquilosante[112]. Además, se ha visto que 14 pacientes del proyecto DECIPHER presentan mutaciones en dicho gen.

Por último, la sobreexpresión del gen de localización nuclear FRG1 se ha asociado con el desarrollo de distrofia muscular facioescapulohumeral al producir splicing alternativo anormal en pre ARN mensajeros[113]. Otro estudio reportó dos bucles de ADN asociados a FRG1 y FRG2 en pacientes con dicha patología, pero no en individuos sanos[114].

En conjunto, las 6 variantes priorizadas presentan evidencia firme asociada a genes que contribuyen al desarrollo de diferentes enfermedades. Además, los fenotipos causados por los genes SRSF10 y FRG1 son producto de anomalías en la maquinaria de splicing de la célula y la patología causada por el gen FRG2C se produce por expresión diferencial en un ARN no-codificante. Ambos procesos pueden verse afectado por variantes en regiones no-codificantes.

Estas variantes tienen que ser analizadas en detalle por un genetista clínico para correlacionar las asociaciones fenotípicas descritas para dichos genes con los hallazgos clínicos en los pacientes. No obstante, el estudio preliminar muestra posibles efectos fenotípicos causados por variantes no-codificantes priorizadas por el clasificador. Son resultados prometedores y esperamos ayuden al diagnostico pacientes con enfermedades raras en el marco del proyecto URUGENOMES.

Chromosoma	Posición	Referencia	Alternativo	Región genómica	Gen	Patogenicidad
1	24.302.049	G	C	Intrónica	SRSF10	0.952
21	9.909.080	G	A	ARN no-codificante	TEKT4P2	0.952
3	75.718.354	G	A	Intergénica	FRG2C	0.952
3	75.718.368	G	A	Intergénica	FRG2C	0.950
3	75.718.374	G	A	Intergénica	FRG2C	0.958
4	190.941.244	G	A	Intergénica	FRG1	0.958

Tabla 11. Variantes no-codificantes priorizadas en el paciente ER23 del proyecto URUGENOMES.

Conclusión

En este proyecto de grado se desarrolló exitosamente un clasificador para priorizar variantes en regiones no codificantes. Para ello se propuso una metodología optimizada para la imputación de características biológicas en variantes genómicas. Se evaluó el desempeño de los clasificadores entrenados sobre diferentes escenarios y se identificó el más informativo para la priorización de variantes patogénicas. Por último, se identificaron dos algoritmos de buen desempeño (RF y GBT). Estos fueron luego entrenados y usados para priorizar variantes patogénicas en regiones no codificantes del genoma de un paciente perteneciente al proyecto URUGENOMES. Sin perjuicio de ello, consideramos que existe espacio para mejorar la performance. Si bien excede el propósito de este trabajo, aparece como una continuación plausible a lo que aquí analizamos.

Agradecimientos

Se le agradece al centro tecnológico Information and Communication Technologies for Verticals (ICT4V) por financiar y hacer posible el presente trabajo de grado.

Bibliografía:

1 - https://www.orpha.net/consor/cgi-bin/Education_AboutRareDiseases.php?lng=EN&stapage=ST_EDUCATION_EDUCATION_ABOUTRAREDISEASES

2 - Van Welly S. y Leufkens H.G.M. (2004), Priority Medicines for Europe and the World "A Public Health Approach to Innovation", 6.19 Rare Diseases. World Health Organization. Disponible en: https://www.who.int/medicines/areas/priority_medicines/Ch6_19Rare.pdf

3 - United States Congress. (2002). Rare Diseases Act of 2002. https://www.gpo.gov/fdsys/pkg/PLAW-107publ280/html/PLAW-107publ280.htm.

4 - The European Parliament and the Council of the European Union (1999). Regulation (EC) No 141/2000 of the European parliament and of the council of 16 December 1999 on orphan medicinal products. http://ec. europa.eu/health//sites/health/files/ files/eudralex/vol-1/reg_2000_141_ cons-2009-07/reg_2000_141_cons-2009- 07_en.pdf.

5 - Viprakasit, V., & Ekwattanakit, S. (2018). Clinical Classification, Screening and Diagnosis for Thalassemia. *Hematology/Oncology Clinics Of North America*, *32*(2), 193-211. doi: 10.1016/j.hoc.2017.11.006

6 - https://www.orpha.net/consor/cgi-bin/Education_AboutOrphanet.php?lng=ES

7 - https://elpais.com/tecnologia/2019/08/12/actualidad/1565610083_028572.html

8 - National Technical Information Services (1989). Report of the National Commission on Orphan Diseases. Office of the Assistant Secretary for Health, Public Health Service, US Department of Health and Human Services, February 1989, p.
17. https://rarediseases.info.nih.gov/files/report_of_the_national_commission_on_orphan_diseases_february_1989.pdf.

9 - Shashi, V. et al. (2013). The utility of the traditional medical genetics diagnostic evaluation in the context of next-generation sequencing for undiagnosed genetic disorders. *Genetics In Medicine*, *16*(2), 176-182. doi: 10.1038/gim.2013.99

10 - https://globalgenes.org/rare-facts/ y https://globalgenes.org/resource-hub/

11 - Graungaard, A., & Skov, L. (2007). Why do we need a diagnosis? A qualitative study of parents? experiences, coping and needs, when the newborn child is severely disabled. *Child: Care, Health And Development*, *33*(3), 296-307. doi: 10.1111/j.1365-2214.2006.00666.x

12 - Dawkins, H.J.S., Draghia-Akli, R., Lasko, P., Lau, L.P.L., Jonker, A.H., Cutillo, C.M., Rath, A., Boycott, K.M., Baynam, G., Lochmu¨ller, H., et al.; International Rare Diseases Research Consortium (IRDiRC) (2018). Progress in Rare Diseases Research 2010-2016: An IRDiRC Perspective. Clin. Transl. Sci. *11*, 11–20.

13 - Austin, C.P., Cutillo, C.M., Lau, L.P.L., Jonker, A.H., Rath, A., Julkowska, D., Thomson, D., Terry, S.F., de Montleau, B., Ardigo` , D., et al.; Interna- tional Rare Diseases Research Consortium (IRDiRC) (2018). Future of Rare Diseases Research 2017-2027: An IRDiRC Perspective. Clin. Transl. Sci. *11*, 21–27.

14 - Ledford, H. (2014). Disease research: Rare insights. *Nature*, *505*(7483), 443-445. doi: 10.1038/nj7483-443a

15 - https://www.rarediseaseday.org/article/what-is-a-rare-disease

16 - Manolio, T. et al. (2009). Finding the missing heritability of complex diseases. *Nature*, *461*(7265), 747-753. doi: 10.1038/nature08494

17 - Maxam, A.M. and Gilbert, W. (1977) A new method for sequencing DNA. Proc. Natl. Acad. Sci. U.S.A. 74, 560–564

18 - Sanger, F. et al. (1977) DNA sequencing with chain-terminating inhibitors. Proc. Natl. Acad. Sci. U.S.A. 74, 5463–5467

19 - van Dijk, E., Auger, H., Jaszczyszyn, Y., & Thermes, C. (2014). Ten years of next-generation sequencing technology. *Trends In Genetics*, *30*(9), 418-426. doi: 10.1016/j.tig.2014.07.001

20 - International Human Genome Sequencing Consortium (2004) Finishing the euchromatic sequence of the human genome. Nature 431, 931–945

21 - Schloss, J.A. (2008) How to get genomes at one ten-thousandth the cost. Nat. Biotechnol. 26, 1113–1115

22 - Auton, A. et al. (2015). A global reference for human genetic variation. *Nature*, *526*(7571), 68-74. doi: 10.1038/nature15393

23 - Koepfli, K., Paten, B., & O'Brien, S. (2015). The Genome 10K Project: A Way Forward. *Annual Review Of Animal Biosciences*, *3*(1), 57-111. doi: 10.1146/annurev-animal-090414-014900

24 - Stephens, Z., Lee, S., Faghri, F., Campbell, R., Zhai, C., & Efron, M. et al. (2015). Big Data: Astronomical or Genomical?. *PLOS Biology*, *13*(7), e1002195. doi: 10.1371/journal.pbio.1002195

25 - Boycott, K., Vanstone, M., Bulman, D., & MacKenzie, A. (2013). Rare-disease genetics in the era of next-generation sequencing: discovery to translation. *Nature Reviews Genetics*, *14*(10), 681-691. doi: 10.1038/nrg3555

26 - Bishop, C. M. (2006), Pattern Recognition and Machine Learning, Springer, ISBN 978-0-387-31073-2

27 - Bucher P. Weight matrix description of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. Journal of Molecular Biology. 1990; 4:563–578.

28 - Heintzman N, et al. Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. Nature Genetics. 2007; 39:311–318. [PubMed: 17277777]

29 - Degroeve S, Baets BD, de Peer YV, Rouz P. Feature subset selection for splice site prediction. Bioinformatics. 2002; 18:S75–S83. [PubMed: 12385987]

30 - Ohler W, Liao C, Niemann H, Rubin GM. Computational analysis of core promoters in the drosophila genome. Genome Biology. 2002; 3

31 - Fraser AG, Marcotte EM. A probabilistic view of gene function. Nature Genetics. 2004; 36:559– 564. [PubMed: 15167932]

32 - Maxwell L. Bileschi, et al. 2019. Using Deep Learning to Annotate the Protein Universe. bioRxiv preprint 626507. doi: 10.1101/626507

33 - Beer MA, Tavazoie S. Predicting gene expression from sequence. Cell. 2004; 117:185–198. [PubMed: 15084257]

34 - Karlic R, R.Chung H, Lasserre J, Vlahovicek K, Vingron M. Histone modification levels are predictive for gene expression. Proceedings of the National Academy of Sciences of the United States of America. 2010; 107:2926–2931. [PubMed: 20133639]

35 - Ioannidis, N., Rothstein, J., Pejaver, V., Middha, S., McDonnell, S., & Baheti, S. et al. (2016). REVEL: An Ensemble Method for Predicting the Pathogenicity of Rare Missense Variants. *The American Journal Of Human Genetics*, *99*(4), 877-885. doi: 10.1016/j.ajhg.2016.08.016

36 - Caron, B., Luo, Y., & Rausell, A. (2019). NCBoost classifies pathogenic non-coding variants in Mendelian diseases through supervised learning on purifying selection signals in humans. *Genome Biology*, *20*(1). doi: 10.1186/s13059-019-1634-2

37 - Smedley, D. et al. (2016). A Whole-Genome Analysis Framework for Effective Identification of Pathogenic Regulatory Variants in Mendelian Disease. *The American Journal of Human Genetics*, *99*, 595-606. doi: 10.1016/j.ajhg.2016.07.005

38 - Rentzsch, P., Witten, D., Cooper, G., Shendure, J., & Kircher, M. (2018). CADD: predicting the deleteriousness of variants throughout the human genome. *Nucleic Acids Research*, *47*(D1), D886-D894. doi: 10.1093/nar/gky1016

39 - Shashi, Vandana et al. (2014). The utility of the traditional medical genetics diagnostic evaluation in the context of next-generation sequencing for undiagnosed genetic disorders. Genetics In Medicine. 16:176. 10.1038/gim.2013.99

40 - Wei, J., Huang, K., Yang, C., & Kang, C. (2016). Non-coding RNAs as regulators in epigenetics. *Oncology Reports*, *37*(1), 3-9. doi: 10.3892/or.2016.5236

41 - Hombach, S., & Kretz, M. (2016). Non-coding RNAs: Classification, Biology and Functioning. *Advances In Experimental Medicine And Biology*, 3-17. doi: 10.1007/978-3-319-42059-2_1

42 - Kovalenko, T., & Patrushev, L. (2018). Pseudogenes as Functionally Significant Elements of the Genome. *Biochemistry (Moscow)*, *83*(11), 1332-1349. doi: 10.1134/s0006297918110044

43 - Santana dos Santos, E., Lallemand, F., Burke, L., Stoppa-Lyonnet, D., Brown, M., Caputo, S., & Rouleau, E. (2018). Non-Coding Variants in BRCA1 and BRCA2 Genes: Potential Impact on Breast and Ovarian Cancer Predisposition. *Cancers*, *10*(11), 453. doi: 10.3390/cancers10110453

44 - Takata, A. (2018). Estimating contribution of rare non-coding variants to neuropsychiatric disorders. *Psychiatry And Clinical Neurosciences*, *73*(1), 2-10. doi: 10.1111/pcn.12774

45 - Hoffman MM, et al. Unsupervised pattern discovery in human chromatin structure through genomic segmentation. Nat Methods. 2012; 9:473–476. [PubMed: 22426492]

46 - Glaab E, Bacardit J, Garibaldi JM, Krasnogor N. Using rule-based machine learning for candidate disease gene prioritization and sample classification of cancer gene expression data. PloS one. 2012; 7:e39932. [PubMed: 22808075]

47 - Ramaswamy S, et al. Multiclass cancer diagnosis using tumor gene expression signatures. Proceedings of the National Academy of Sciences of the United States of America. 2001; 98:15149–54. [PubMed: 11742071]

48 - Quang, D., Chen, Y., & Xie, X. (2014). DANN: a deep learning approach for annotating the pathogenicity of genetic variants. *Bioinformatics*, *31*(5), 761-763. doi: 10.1093/bioinformatics/btu703

49 - Zhang, S., He, Y., Liu, H., Zhai, H., Huang, D., & Yi, X. et al. (2019). regBase: whole genome base-wise aggregation and functional prediction for human non-coding regulatory variants. *Nucleic Acids Research*. doi: 10.1093/nar/gkz774

50 - Adzhubei, I., Jordan, D., & Sunyaev, S. (2013). Predicting Functional Effect of Human Missense Mutations Using PolyPhen-2. *Current Protocols In Human Genetics*, *76*(1), 7.20.1-7.20.41. doi: 10.1002/0471142905.hg0720s76

51 - Ng, P. (2003). SIFT: predicting amino acid changes that affect protein function. *Nucleic Acids Research*, *31*(13), 3812-3814. doi: 10.1093/nar/gkg509

52 - Choi, Y., Sims, G., Murphy, S., Miller, J., & Chan, A. (2012). Predicting the Functional Effect of Amino Acid Substitutions and Indels. *Plos ONE*, *7*(10), e46688. doi: 10.1371/journal.pone.0046688

53 - https://dx29.ai/clinician

54 - https://elpais.com/tecnologia/2019/08/12/actualidad/1565610083_028572.html

55 - The ENCODE Project Consortium, An integrated encyclopedia of DNA elements in the human genome. (2012). *Nature*, *489*(7414), 57-74. doi: 10.1038/nature11247

56 - Kim, J., Hu, C., Moufawad El Achkar, C., Black, L., Douville, J., & Larson, A. et al. (2019). Patient-Customized Oligonucleotide Therapy for a Rare Genetic Disease. *New England Journal Of Medicine*, *381*(17), 1644-1652. doi: 10.1056/nejmoa1813279

57 - Kolata, G. (2020). Scientists Designed a Drug for Just One Patient. Her Name Is Mila. Retrieved 23 May 2020, from https://www.nytimes.com/2019/10/09/health/mila-makovec-drug.html

58 - Benson, D., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Lipman, D., Ostell, J., & Sayers, E. (2012). GenBank. *Nucleic Acids Research*, *41*(D1), D36-D42. doi: 10.1093/nar/gks1195

59 - El-Gebali, S., Mistry, J., Bateman, A., Eddy, S., Luciani, A., & Potter, S. et al. (2018). The Pfam protein families database in 2019. *Nucleic Acids Research*, *47*(D1), D427-D432. doi: 10.1093/nar/gky995

60 - International HapMap Constortium. A second generation human haplotype map of over 3.1 million SNPs. *Nature*, 449:851-862. 2007

61 - Bycroft, C., Freeman, C., Petkova, D., Band, G., Elliott, L., & Sharp, K. et al. (2018). The UK Biobank resource with deep phenotyping and genomic data. *Nature*, *562*(7726), 203-209. doi: 10.1038/s41586-018-0579-z

62 - A global reference for human genetic variation, The 1000 Genomes Project Consortium, Nature 526, 68-74 (01 October 2015) doi:10.1038/nature15393

63 - Stenson PD, Mort M, Ball EV, Evans K, Hayden M, Heywood S, Hussain M, Phillips AD, Cooper DN. The Human Gene Mutation Database: towards a comprehensive repository of inherited mutation data for medical research, genetic diagnosis and next-generation sequencing studies. *Hum Genet* 136:665-677, 2017.

64 - Chen R, Shi L, Hakenberg J, Naughton B, Sklar P. et al. *Show more authors.* "Analysis of 589,306 genomes identifies individuals resilient to severe Mendelian childhood diseases." Nature Biotechnology. 11 April 2016

65 - National Center for Biotechnology Information (NCBI)[Internet]. Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information; [1988] – [cited 2017 Apr 06]. Available from: https://www.ncbi.nlm.nih.gov/

66 - Harte, N., Silventoinen, V., Quevillon, E., Robinson, S., Kallio, K., & Fustero, X. et al. (2004). Public web-based services from the European Bioinformatics Institute. *Nucleic Acids Research*, *32*(Web Server), W3-W9. doi: 10.1093/nar/gkh405

67 - HARLAND, S. (1960). The National Institute of Genetics, Japan. *Nature*, *186*(4721), 286-286. doi: 10.1038/186286a0

68 - Davydov E.V. Goode D.L. Sirota M. Cooper G.M. Sidow A. Batzoglou S. (2010) Identifying a high fraction of the human genome to be under selective constraint using GERP++. *PLoS Comput. Biol.* , 6, e1001025.

69 - Karczewski, K., Francioli, L., Tiao, G., Cummings, B., Alföldi, J., & Wang, Q. et al. (2019). The mutational constraint spectrum quantified from variation in 141,456 humans. doi: 10.1101/531210

70 - Online Mendelian Inheritance in Man, OMIM®. McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University (Baltimore, MD), {date}. World Wide Web URL: https://omim.org/

71 - Richards, S., Aziz, N., Bale, S., Bick, D., Das, S., & Gastier-Foster, J. et al. (2015). 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. *Genetics In Medicine*, *17*(5), 405-423. doi: 10.1038/gim.2015.30

72 - https://www.displayr.com/different-types-of-missing-data/

73 - Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, *25*(14), 1754-1760. doi: 10.1093/bioinformatics/btp324

74 - Wang K, Li M, Hakonarson H. ANNOVAR: Functional annotation of genetic variants from next-generation sequencing data *Nucleic Acids Research*, 38:e164, 2010

75 - Ngak-Leng Sim, Prateek Kumar, Jing Hu, Steven Henikoff, Georg Schneider, Pauline C. Ng, SIFT web server: predicting effects of amino acid substitutions on proteins, *Nucleic Acids Research*, Volume 40, Issue W1, 1 July 2012, Pages W452–W457, https://doi.org/10.1093/nar/gks539

76 - Adzhubei IA, Schmidt S, Peshkin L, Ramensky VE, Gerasimova A, Bork P, Kondrashov AS, Sunyaev SR. Nat Methods **7**(4):248-249 (2010).

77 - Chengliang Dong, Peng Wei, Xueqiu Jian, Richard Gibbs, Eric Boerwinkle, Kai Wang, Xiaoming Liu, Comparison and integration of deleteriousness prediction methods for nonsynonymous SNVs in whole exome sequencing studies, *Human Molecular Genetics*, Volume 24, Issue 8, 15 April 2015, Pages 2125–2137, https://doi.org/10.1093/hmg/ddu733

78 - Chun, S., & Fay, J. (2009). Identification of deleterious mutations within three human genomes. *Genome Research*, *19*(9), 1553-1561. doi: 10.1101/gr.092619.109

79 - Shihab,H.A. et al. (2013) Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden Markov models. Hum. Mutat., 34, 57–65.

80 - Ng,P.C. and Henikoff,S. (2002) Accounting for human polymorphisms predicted to affect protein function. Genome Res., 12, 436–446.

81 - Capriotti, E., Calabrese, R. & Casadio, R. *Bioinformatics* **22**, 2729–2734 (2006).

82 - Stenson PD, Mort M, Ball EV, Howells K, Phillips AD, Thomas NS, Cooper DN. 2009. The Human Gene Mutation Database: 2008 update. Genome Med 1:13.

83 - Apweiler R, Bairoch A, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, et al. 2004. UniProt: the Universal Protein knowledgebase. Nucleic Acids Res 32:D115–119

84 - Choi, Y., Sims, G., Murphy, S., Miller, J., & Chan, A. (2012). Predicting the Functional Effect of Amino Acid Substitutions and Indels. *Plos ONE*, *7*(10), e46688. doi: 10.1371/journal.pone.0046688

85 - Carter H, Douville C, Yeo G, Stenson PD, Cooper DN, Karchin R (2013) Identifying Mendelian disease genes with the Variant Effect Scoring Tool BMC Genomics. 14(3) 1-16.

86 - Schwarz J.M. Rodelsperger C. Schuelke M. Seelow D. (2010) MutationTaster evaluates disease-causing potential of sequence alterations. *Nat. Methods* , 7, 575–576.

87 - Reva B. Antipin Y. Sander C. (2011) Predicting the functional impact of protein mutations: application to cancer genomics. *Nucleic Acids Res.* , 39, e118.

88 - Kircher M. Witten D.M. Jain P. O'Roak B.J. Cooper G.M. Shendure J. (2014) A general framework for estimating the relative pathogenicity of human genetic variants. *Nat. Genet.* , 46, 310–315.

89 - Davydov E.V. Goode D.L. Sirota M. Cooper G.M. Sidow A. Batzoglou S. (2010) Identifying a high fraction of the human genome to be under selective constraint using GERP++. *PLoS Comput. Biol.* , 6, e1001025.

90 - Garber M. Guttman M. Clamp M. Zody M.C. Friedman N. Xie X. (2009) Identifying novel constrained elements by exploiting biased substitution patterns. *Bioinformatics* , 25, i54–i62.

91 -
Cooper G.M. Stone E.A. Asimenos G. Program N.C.S. Green E.D. Batzoglou S. Sidow A. (2005) Distributio n and intensity of constraint in mammalian genomic sequence. *Genome Res.* , 15, 901–913.

92 - Shihab, H., Rogers, M., Gough, J., Mort, M., Cooper, D., & Day, I. et al. (2015). An integrative approach to predicting the functional effects of non-coding and coding sequence variation. *Bioinformatics*, *31*(10), 1536-1543. doi: 10.1093/bioinformatics/btv009

93 - Gulko, B., Hubisz, M., Gronau, I., & Siepel, A. (2015). A method for calculating probabilities of fitness consequences for point mutations across the human genome. *Nature Genetics*, *47*(3), 276-283. doi: 10.1038/ng.3196

94 - Lek, M., Karczewski, K., Minikel, E., Samocha, K., Banks, E., & Fennell, T. et al. (2016). Analysis of protein-coding genetic variation in 60,706 humans. *Nature*, *536*(7616), 285-291. doi: 10.1038/nature19057

95 - Karczewski, K., Francioli, L., Tiao, G., Cummings, B., Alföldi, J., & Wang, Q. et al. (2019). Variation across 141,456 human exomes and genomes reveals the spectrum of loss-of-function intolerance across human protein-coding genes. doi: 10.1101/531210

96 - Glusman G, Caballero J, Mauldin DE, Hood L and Roach J (2011) KAVIAR: an accessible system for testing SNV novelty. Bioinformatics, doi: 10.1093/bioinformatics/btr540

97 - A. Siepel and D. Haussler, 2005. Phylogenetic hidden Markov models. In R. Nielsen, ed., *Statistical Methods in Molecular Evolution*, pp. 325-351. Springer, New York.

98 - Teng, M., Ichikawa, S., Padgett, L., Wang, Y., Mort, M., & Cooper, D. et al. (2012). regSNPs: a strategy for prioritizing regulatory single nucleotide substitutions. *Bioinformatics*, *28*(14), 1879-1886. doi: 10.1093/bioinformatics/bts275

99 - Li, Q., & Wang, K. (2017). InterVar: Clinical Interpretation of Genetic Variants by the 2015 ACMG-AMP Guidelines. *The American Journal Of Human Genetics*, *100*(2), 267-280. doi: 10.1016/j.ajhg.2017.01.004

100 - Ritchie, G., Dunham, I., Zeggini, E., & Flicek, P. (2014). Functional annotation of noncoding sequence variants. *Nature Methods*, *11*(3), 294-296. doi: 10.1038/nmeth.2832

101 - Ionita-Laza, I., McCallum, K., Xu, B., & Buxbaum, J. (2016). A spectral approach integrating functional genomic annotations for coding and noncoding variants. *Nature Genetics*, *48*(2), 214-220. doi: 10.1038/ng.3477

102 - Data Mining with R, learning with case studies, Luis Torgo, CRC Press 2010

103 - van Buuren, Stef. Flexible Imputation of Missing Data. Chapman & Hall/CRC, Boca Raton FL, 2012.

104 - Yu-Sung Su, Andrew Gelman, Jennifer Hill, Masanao Yajima. (2011). Multiple Imputation with Diagnostics (mi) in R: Opening Windows into the Black Box. *Journal of Statistical Software* 45(2).

105 - Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations inR. *Journal Of Statistical Software*, *45*(3). doi: 10.18637/jss.v045.i03

106 - Gary King, James Honaker, Anne Joseph, Kenneth Scheve. "Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation", American POlitical Science Review, Vol. 95, No. 1 (March, 2001): Pp. 49-69.

107 - Altman, N. S. (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". *The American Statistician*. 46 (3): 175–185.

108 - Breiman, Leo (2001). «Random Forests». *Machine Learning* 45 (1): 5–32.

109 - Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome(2008). *The Elements of Statistical Learning* (2nd ed.). Springer.

110 - Shin, C., Feng, Y., Manley, J. L. Dephosphorylated SRp38 acts as a splicing repressor in response to heat shock. Nature 427: 553-558, 2004.

111 - DECIPHER: Database of Chromosomal Imbalance and Phenotype in Humans using Ensembl Resources. Firth, H.V. *et al.*, 2009. Am.J.Hum.Genet 84, 524-533
(DOI: dx.doi.org/10/1016/j.ajhg.2009.03.010)

112 - Xie Z, Li J, Wang P, et al. Differential Expression Profiles of Long Noncoding RNA and mRNA of Osteogenically Differentiated Mesenchymal Stem Cells in Ankylosing Spondylitis. *J Rheumatol*. 2016;43(8):1523-1531. doi:10.3899/jrheum.151181

113 - Gabellini, D., D'Antona, G., Moggio, M., Prelle, A., Zecca, C., Adami, R., Angeletti, B., Ciscato, P., Pellegrino, M. A., Bottinelli, R., Green, M. R., Tupler, R. Facioscapulohumeral muscular dystrophy in mice overexpressing FRG1. Nature 439: 973-977, 2006.

114 - Petrov, A., Pirozhkova, I., Carnac, G., Laoudj, D., Lipinski, M., Vassetzky, Y. S. Chromatin loop domain organization within the 4q35 locus in facioscapulohumeral dystrophy patients versus normal human myoblasts. Proc. Nat. Acad. Sci. 103: 6982-6987, 2006.

Anexo

A

Regresión Lineal

	Significancia	Patogenicidad por niveles	Patogénica vs Benigna
Intercepto	TRUE	TRUE	TRUE

B

Árboles de decisión

	Significancia	Patogenicidad por niveles	Patogénica vs Benigna
Confianza	0.01	0.01	0.1325
Instancias por hoja	2	5	4

C

Vecinos más cercanos

	Significancia	Patogenicidad por niveles	Patogénica vs Benigna
Vecinos	9	13	5

D

Bosques Aleatorios

	Significancia	Patogenicidad por niveles	Patogénica vs Benigna
Variables tomadas por árbol	2	3	2
N° de árboles	500	500	1600

E

Gradient Boosted Trees

	Significancia	Patogenicidad por niveles	Patogénica vs Benigna
Iteraciones	50	50	400
Profundidad por árbol	1	1	5
Encojimiento	0.4	0.4	0.2
Perdida mínima	0	0	0
Variables tomadas por árbol (%)	0.6	0.8	0.4
Suma mínima de los pesos	1	1	1
Muestreo	0.5	0.5	0.8

F

Maquinas de Soporte de Vectores

	Significancia	Patogenicidad por niveles	Patogénica vs Benigna
Costo	0.25	0.25	0.25
Grado	1	1	3
Escalado	0.001	0.001	1
Sigma	0.190	0.233	0.276

G

Redes Neuronales

	Significancia	Patogenicidad por niveles	Patogénica vs Benigna
Capas ocultas	3	3	3
Función de activación	relu	relu	relu
Ratio de aprendizaje	0.01	0.01	0.01
Neuronas por capa	2500	5000	4000
Iteraciones	74	88	53

Tabla suplementaria 1. Parámetros utilizados para entrenar los modelos de Regresión Lineal, Árboles de Decisión, Vecinos más cercanos, Bosques Aleatorios, Gradient Boosted Trees, Maquinas de Soporte de Vectores y Redes Neuronales (A, B, C, D, E, F y G respectivamente).