Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Predicción de patogenicidad en SNPs usando aprendizaje automático

Tesis de Licenciatura en Ciencias de la Computación

Martín Ezequiel Langberg

Director: Ariel Berenstein
Codirector: Pablo Turjanski
Buenos Aires, 2019

PREDICCIÓN DE PATOGENICIDAD EN POLIMORFISMOS DE UN SOLO NUCLEÓTIDO USANDO APRENDIZAJE AUTOMÁTICO

El estudio de enfermedades de origen genético ha tenido un desarrollo constante y acelerado en los últimos años en parte gracias a nuevas técnicas de secuenciación del genoma, que permiten el análisis del material genético de pacientes a nivel de exomas y genomas completos con costos cada vez más reducidos y accesibles. En este contexto, resulta de gran importancia la capacidad de identificar polimorfismos de un solo nucleótido (SNPs, por sus siglas en inglés) causales de enfermedades humanas y diferenciarlos respecto de aquellos con efecto inocuo para el organismo. Dada la gran cantidad de SNPs presentes en el genoma humano, esta línea de investigación ha cobrado un marcado interés por parte de la comunidad científica en general, motivando esfuerzos interdisciplinarios, en particular de trabajos que subyacen en la frontera de las ciencias de la computación y las ciencias biológicas.

En el presente trabajo, mediante el uso de técnicas de aprendizaje automático supervisado convencionales hemos elaborado un método de predicción de patogenicidad para SNPs en regiones codificantes que resulten en un cambio de aminoácido, normalmente referidas como SNPs con cambio de sentido.

Nuestro modelo de clasificación binaria, se basa en las fuentes de Clinvar y Humsavar para clasificar el efecto patogénico de SNPs conocidos, y en distintas fuentes de información para extraer variables que caractericen los SNPs desde distintas aristas biológicas.

En particular hemos explorado la importancia relativa y el poder predictivo de variables que den a cuenta del cambio estructural producido por el cambio de aminoácido (variación de energía, superficie de exposición del aminoácido, entre otras), variables de tipo físico-químico (hidrofobicidad, aromaticidad, polaridad, etc) y de conservación a nivel genómico (PhyloP y PhastCons, por ejemplo). Evaluamos la importancia relativa de cada una de estas dimensiones aplicando técnicas clásicas de aprendizaje automático supervisado: Regresión Logística, Support Vector Machines y Random Forest. Finalmente, evaluamos la combinación de las variables con una técnica más avanzada de aprendizaje automático, XGBoost, con el que alcanzamos un AUC de 0.90.

Palabras claves: Aprendizaje Automático, Bioinformática, SNPs, Patogenicidad, Genética.

AGRADECIMIENTOS

Haber terminado esta Tesis de Licenciatura no solamente significó la culminación de un arduo período de trabajo sino que también representa el "cierre" formal de una etapa. Esto me invita a la reflexión sobre los momentos transitados durante todos estos años.

Este trabajo no hubiera podido realizarse sin el acompañamiento recibido de muchas personas en diferentes momentos, y quiero aprovechar esta oportunidad para agradecer a todas aquellas que mi memoria me ayuda a recordar.

A mi familia: Mis padres, hermanos, tíos, primos y abuelos. Gracias por el apoyo y el amor recibido. Especialmente a mi abuela Nelly y a mi abuela Rechel, que ya no están. Fueron mis primeras profesoras y el amor infinito que me dieron es algo que me acompaña y me da fuerzas todos los días. Se que se alegrarían muchísimo por este logro.

A mi primer grupo de amigos durante la cursada del CBC: A Lucas y a Diego. Recuerdo con mucha felicidad haberlos tenido como compañeros en ese primer paso. Si bien después nuestros caminos se separaron, fue muy importante ese primer año de temores y éxitos compartidos.

A mis amigos eternos del secundario: Javier y Nico. Horas y horas de batallas épicas, y de campeonatos mundiales. Juntarme con ustedes siempre es el escape perfecto a otros mundos.

A mis compañeros Mezzeteros: Ari, Javi y Gabi. Fue muy divertido haber compartido tantas materias y trabajos prácticos, pero lo mejor fue nuestro primer trabajo. Cuando llegamos a la entrevista nos confundieron con una banda de rock, y lo que siguió después fue igual de bizarro. Seguimos riéndonos de ese momento en que soñamos con ser los nuevos rockstars de la tecnología.

A mis queridos Pepes: Guille, Fixman, Luigi, Diego y JP. Estoy muy feliz y orgulloso de haber aprendido con ustedes y sobre todo de ustedes. Cómo olvidar las incontables noches comiendo pizza en el bar de deportes. Se que tengo un hogar en cualquier lugar del mundo en donde se encuentren. Quiero recordar a Leopoldo también, por haberme dado un empujón de confianza en un par de momentos importantes. Gracias, Leo.

A mis amigos del trabajo: Maxi, Yani, Pedro y Nico. Son una gran razón de mis risas durante el día a día.

A mi gran amigo y hermano del alma Dami: Por tantas caminatas random, por los viajes y por estar siempre.

También quiero agradecer a mis directores Pablo y Ariel por todas las juntadas, por las ideas y la buena onda. En especial a Ariel por haber sido el ideólogo oficial detrás de esta "fuga" universitaria y en especial a Pablo por sus Vauquitas. Por último pero no menos importante, a la educación pública y a sus docentes por haberme acogido en cada de mis etapas de aprendizaje, me siento privilegiado y en deuda por haberme enseñado no solamente conceptos técnicos, también me ayudó a tener una conciencia social y un pensamiento crítico que me va a acompañar toda la vida.

A mis maestros de la vida: Marta, Jorge, y José.

Índice general

1. INTRODUCCIÓN

1.1. Problema biológico

La biología, y en particular la genética, han tenido un lugar destacado en el siglo XX gracias a enormes avances como el descubrimiento de la estructura molecular del ADN, con el aporte de Franklin [1]; y Crick y Watson [2]. A partir de esos hitos científicos, se sucedieron grandes esfuerzos a nivel internacional, como el Proyecto Genoma Humano [3] o ENCODE [4], que tienen como objetivo general el mapeo entre código genético humano y sus elementos funcionales. Estos trabajos, sumados a los recientes avances en las tecnologías de secuenciación masiva, han generado un creciente volumen de datos genómicos que a su vez favorecieron la aparición de nuevos tipos de tratamientos en el área de la medicina personalizada [5].

La medicina personalizada o de precisión es un modelo médico en donde las intervenciones se realizan a individuos o a grupos en riesgo a diferencia del modelo tradicional que apunta a la población en general. **Dentro de este área, uno de los objetivos críticos que se busca resolver es la detección de variaciones genéticas que deriven en enfermedades**.

1.1.1. El dogma central de la biología

El genoma humano está compuesto por el ADN (ácido desoxirribonucleico), que se encuentra en los 23 pares de cromosomas (moléculas de ADN) y en las mitocondrias, en las que existe una molécula de ADN denominada ADN mitocondrial. El ADN se compone de 4 bases: Adenina, Timina, Citosina, y Guanina. Estas bases se representan con su letra inicial: A, T, C y G. Cada una de estas bases está unida a otra (Adenina-Timina y Citosina-Guanina) formando pares de bases en forma helicoidal. A su vez estas cadenas pueden ser divididas en genes, que son regiones del ADN que codifican funciones. El conjunto de genes de un organismo es lo que se conoce como el genoma.

Para entender un poco cómo funciona el proceso por el cual la información genética se expresa en el organismo recurrimos al llamado "Dogma central de la biología", esbozado por Francis Crick en 1958 [6] y revisado en 1970 [7]. Este dogma, si bien fue ampliado a lo largo del tiempo, explica de forma general este proceso. Como vemos en la figura 1.1, el ADN (ácido desoxirribonucleico) se transcribe en forma de ARN para luego formar una proteína. El ARN (ácido ribonucleico), es una mólecula polimérica, al igual que el ADN, y también posee las mismas bases que el ADN, excepto la timina (T) que se reemplaza por el uracilo (U). Finalmente, el ARN es utilizado para unir los aminoácidos que forman la secuencia proteica, en un proceso denominado Traducción. La secuencia del ARN se compone de tripletes adyacentes de bases que se denominan codones, y cada aminoácido está codificado por uno de estos codones (ver figura 1.2).

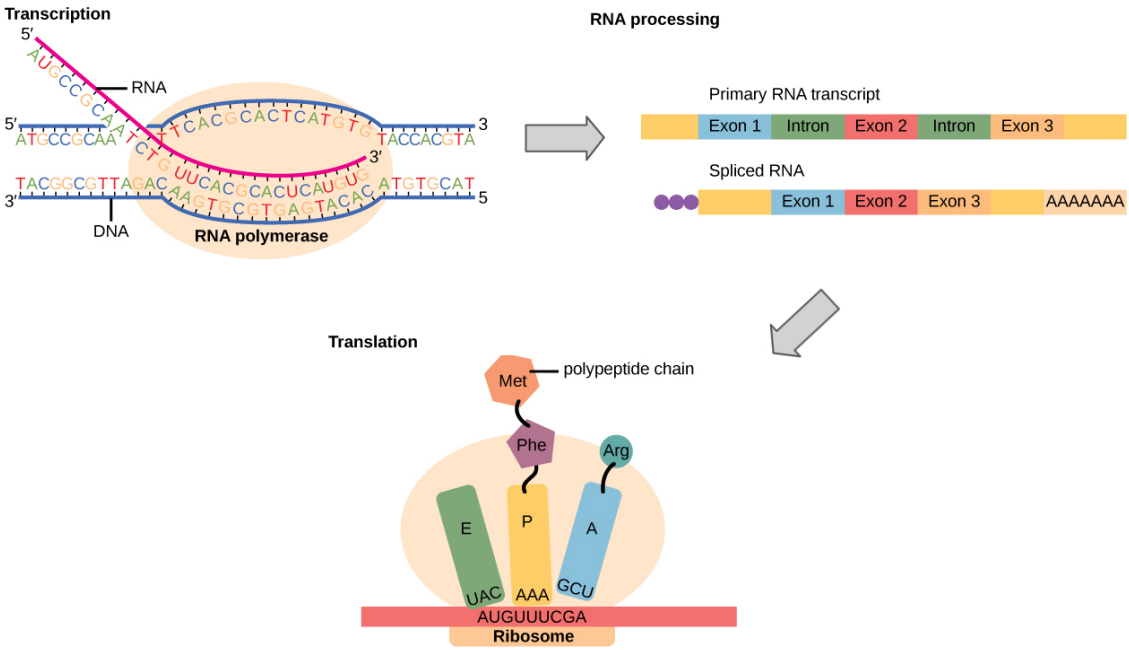


Fig. 1.1: Transferencia de la información genética. El ADN es transcripto en forma de ARN (ARN mensajero). Luego el ARN mensajero es ensamblado uniendo los exones (partes codificantes del gen) en un proceso denominado *splicing*. Por último, los ribosomas usan la información del ARN mensajero para unir los aminoácidos en forma de proteína. Imagen tomada de OpenStax [8].

Second letter

First letter	U	C	A	G	Third letter
U	UUU Phe UUC Phe UUA Leu UUG Leu	UCU Ser UCC Ser UCA Ser UCG Ser	UAU Tyr UAC Tyr **UAA Stop** **UAG Stop**	UGU Cys UGC Cys **UGA Stop** UGG Trp	U C A G
C	CUU Leu CUC Leu CUA Leu CUG Leu	CCU Pro CCC Pro CCA Pro CCG Pro	CAU His CAC His CAA Gln CAG Gln	CGU Arg CGC Arg CGA Arg CGG Arg	U C A G
A	AUU Ile AUC Ile AUA Ile **AUG Met**	ACU Thr ACC Thr ACA Thr ACG Thr	AAU Asn AAC Asn AAA Lys AAG Lys	AGU Ser AGC Ser AGA Arg AGG Arg	U C A G
G	GUU Val GUC Val GUA Val GUG Val	GCU Ala GCC Ala GCA Ala GCG Ala	GAU Asp GAC Asp GAA Glu GAG Glu	GGU Gly GGC Gly GGA Gly GGG Gly	U C A G

Fig. 1.2: Tabla de codones para generar un determinado aminoácido o símbolo de terminación (*Stop*). Imagen tomada de OpenStax [8].

1.1.2. Variaciones genéticas

Las variaciones genéticas son diferencias en el ADN entre individuos de una población [9]. Estas variaciones pueden ser causadas por mutaciones en el momento de la replicación del material genético, pudiendo ser de carácter permanente. Las mutaciones genéticas suelen representar un porcentaje muy pequeño con respecto a la secuencia completa del genoma (alrededor del 0.5 %), pero muchas de ellas suelen ser responsables de variaciones fenotípicas, es decir, nuestros rasgos "observables".

Estas variaciones se pueden dividir en tres grupos principales:

- Polimorfismos de un sólo nucleótido (SNPs): sustitución de un único par de bases.
- Inserciones o deleciones (indels): pueden ocurrir en un intervalo grande del ADN de entre 2 a 200 pares de bases.
- Variaciones estructurales: ocurren en secuencias largas de bases, y pueden ser indels, inversiones, duplicaciones, entre otras.

Finalmente, cualquiera de estas variaciones pueden ser beneficiosas para el organismo, neutrales (sin efecto alguno) o perjudiciales.

1.1.3. Polimorfismos de un sólo nucleótido (SNPs)

En el marco de esta tesis estudiaremos los polimorfismos de un sólo nucleótido, o SNPs por sus siglas en inglés (Single Nucleotide Polymorphism). Una persona posee en promedio alrededor de 4 a 5 millones de SNPs. Mientras la mayoría de ellos no tiene efecto en su desarrollo, algunos de ellos pueden variar la respuesta a ciertas drogas, o el riesgo de sufrir algunas enfermedades.

En la figura 1.3 podemos ver los distintos tipos de SNP. De acuerdo al lugar, pueden ocurrir en una zona codificante, es decir una porción del gen que codifica una proteína, o en una zona no codificante, como los intrones (partes del gen no codificante).

Dentro de las sustituciones en la zona codificante, algunas mutaciones pueden ser sinónimas, es decir que existe un cambio en uno de los nucleótidos del ADN pero no genera un cambio en el aminoácido que codifica. Podemos ver en la tabla 1.2 que muchos codones codifican el mismo aminoácido, por ejemplo, los codones AUU, AUC y AUA codifican el aminoácido isoleucina. Luego, si en la cadena de ADN, el codón AUA sufre una mutación en su último nucleótido, pasando a ser AUC, dicho codón seguirá codificando para el mismo aminoácido. Otro tipo de sustituciones, denominadas mutaciones sin sentido (*nonsense*) codifican un codón de terminación (*stop*), que resulta en un fin de codificación prematuro y en general una proteína no funcional.

En particular estudiaremos los SNPs con cambio de sentido (*missense*), es decir aquellos que deriven en un cambio de aminoácido en la proteína producida por la variante del gen.

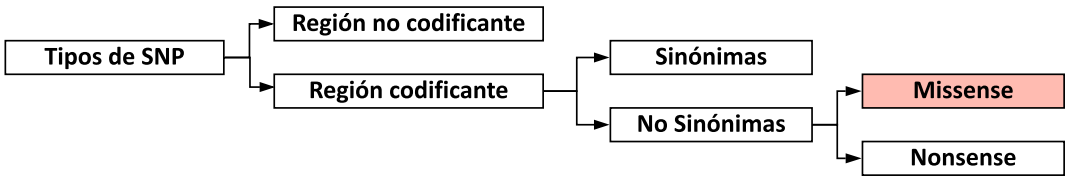


Fig. 1.3: Diferentes tipos de SNP de acuerdo a su posición en el genoma y su efecto. En este trabajo nos concentraremos únicamente en las sustituciones *missense*.

1.1.4. Bases de datos *ómicas*

En la actualidad existen diferentes bases de datos públicas (dbSNP, SNPedia, HapMap, entre otras) que registran millones de estos SNPs *missense*. Los costos de una secuenciación completa siguen bajando de forma acelerada desde el Proyecto Genoma Humano [10] (ver figura 1.4). Esto ha permitido generar un número creciente de estudios que permiten asociar polimorfismos genéticos a enfermedades (ver figura 1.5). Diferentes bases, como Humsavar [11] o Clinvar [12], contienen reportes curados con dichos resultados, que se actualizan periódicamente y pueden variar con nueva evidencia. Sin embargo, todavía existe un gran número de SNPs *missense* de los cuales se desconoce su efecto en el organismo.

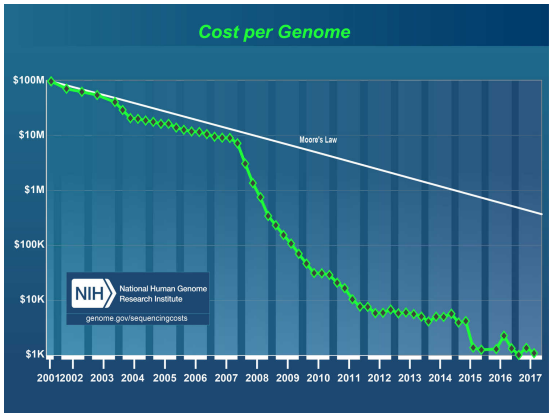


Fig. 1.4: Costo por secuenciación del genoma (NCBI-NIH, 2017). La curva verde corresponde al costo en dólares y la curva blanca equivale a la curva de la ley de Moore, es decir, si se reduciera a la mitad cada año.

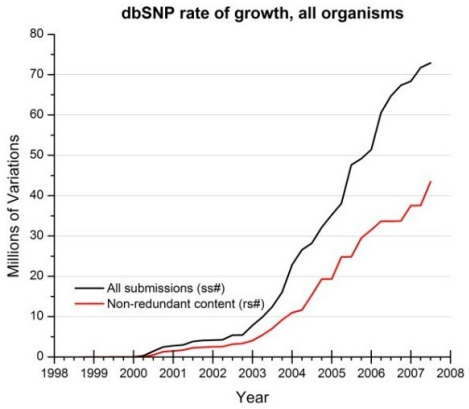


Fig. 1.5: Crecimiento de la base dbSNP a partir del Proyecto Genoma Humano (NCBI-NIH, 2008). La curva negra corresponde a todos los SNPs subidos a la base y la curva roja a los clusters de SNPs que referencian a la misma posición del genoma.

El problema biológico que intentaremos atacar es poder predecir aquellos SNPs missense no investigados aún cuyo cambio en el aminoácido de la proteína generada pueda estar asociado a alguna patogénesis.

1.2. Enfoque computacional

Para abordar este problema, decidimos usar métodos de aprendizaje automático. El aprendizaje automático es un método computacional (dentro del área de la inteligencia artificial y la estadística) que consiste en aprender a partir de los datos.

Existen diferentes formas de categorizar los algoritmos de aprendizaje automático. De acuerdo a Barber [13], una de las formas posibles es la categorización de acuerdo a la cantidad de tipo de supervisión durante la fase de entrenamiento. Este aprendizaje puede ser:

- Supervisado, en donde se utilizan ejemplos previos que están "rotulados", es decir que poseen una variable de respuesta conocida y se busca conseguir una predicción lo más certera posible sobre nuevos datos. Esta variable de respuesta puede ser continua (problema de regresión) o dividida en clases (problema de clasificación).

- No supervisado, donde el objetivo es encontrar distintos grupos o clusters dentro de los datos.

- Semi-supervisado, en el que se trabaja con un pequeño conjunto de datos etiquetados y un conjunto mucho mayor de datos no etiquetados. El objetivo consiste en usar este último conjunto para mejorar el clasificador construido con los datos etiquetados.

- Por refuerzo, donde un *agente* observa el entorno y realiza diferentes acciones para maximizar la recompensa. El modelo a aprender entonces es la estrategia que le permite tomar decisiones ante determinadas situaciones.

Durante la fase de entrenamiento estos algoritmos ajustan sus parámetros de acuerdo a los datos recibidos. En la mayoría de los casos estos algoritmos también poseen hiperparámetros, que no se modifican durante la fase de entrenamiento y determinan algunas de sus características. Estos hiperparámetros pueden ser optimizados mediante técnicas de validación cruzada (ver sección 1.2.2).

La creciente producción de trabajos nos aporta una gran cantidad de efectos conocidos de las variantes proteicas, y a su vez la ubicación (*locus*) del SNP *missense* responsable en el genoma. Estos datos se encuentran (en gran parte) de forma abierta y gratuita, lo que nos permite aplicar este enfoque computacional. Usando esta fuente de datos entrenaremos un modelo (de forma supervisada) que pueda predecir con un cierto grado de precisión el efecto de una variante aún no reportada.

1.2.1. Aprendizaje supervisado

Como mencionamos anteriormente, el aprendizaje supervisado trata de predecir una respuesta usando un modelo generado con datos correctamente etiquetados. Definido de manera formal, dado un set de datos $\mathcal{D} = \{(x^n, y^n), n = 1...N\}$ buscamos aprender la relación entre el ejemplo x y la variable de respuesta y tal que al recibir un nuevo ejemplo x^* la respuesta predicha y^* sea precisa [13]. La precisión está definida formalmente por la función de pérdida o *Loss Function*, $L(y^{pred}, y^{true})$. Esta función nos permite medir el costo de errar en la predicción y por lo tanto entrenar nuestro modelo de manera que el valor de la función de pérdida usando los datos de entrenamiento sea mínimo o cercano al mínimo.

En el contexto de nuestro problema, cada vector x del set de datos $\mathcal{D}$ es un conjunto de variables que describen a distintos niveles (estructural, físico-químico, genómico) al SNP y a su variante proteica producida e y es el efecto producido en el organismo, basándonos en reportes de Humsavar y Clinvar. Al ser una variable de tipo binaria, podemos aplicar algoritmos de clasificación para este problema. Posteriormente en la secciones 1.2.2, 1.2.3 y 1.2.4 analizaremos el proceso general por el cuál estos algoritmos son entrenados y las métricas para evaluar su desempeño.

A continuación haremos un recorrido por los distintos métodos de clasificación que utilizaremos en esta tesis. Estos métodos se encuentran implementados en el módulo `scikit-learn` de Python [14], excepto por el método Gradient Boosting y su implementación XGBoost [15]. En cada uno de los apartados mencionaremos su funcionamiento general, sus parámetros y sus hiperparámetros. Para estos últimos mencionaremos entre paréntesis su nombre usado en la implementación.

Regresión logística

La regresión logística (LR) es un modelo basado en la regresión lineal. Al igual que ésta, consiste en buscar los coeficientes de una función de manera que el valor de la función de pérdida sea el mínimo. A diferencia de la regresión lineal, que usa una función lineal (o polinomial) para aproximar los puntos (que son valores continuos), la regresión logística usa la función logística para aproximar los valores (que son categóricos). Esta función, generalizada para múltiples variables predictoras y variable de respuesta binaria se define como:

$$h_\theta(\boldsymbol{X}) = \frac{1}{1 + e^{\theta^T \boldsymbol{X}}} = Pr(Y = 1|\boldsymbol{X}; \theta)$$

donde $\boldsymbol{X}$ representa el vector de variables predictoras, θ es el vector de coeficientes (θ^T representa al vector transpuesto) e Y es la variable de respuesta [13]. Lo que se busca modelar, es la probabilidad de pertenecer a una clase determinada (simbolizada con '1'), cuando las variables observadas son $\boldsymbol{X}$ y usando los parámetros θ. La probabilidad de pertenecer a la otra clase entonces, es igual a 1 - $h_\theta(\boldsymbol{X})$.

Los parámetros θ son obtenidos buscando maximizar la verosimilitud con los parámetros de la distribución real de los datos [16], mientras que los hiperparámetros ($\tilde{\theta}$) se obtienen usando validación cruzada (ver sección 1.2.2). Los hiperparámetros que buscaremos optimizar son dos:

- El parámetro de regularización (`C`): Parámetro usado para penalizar el uso de muchas variables en la función de pérdida, simplificando el modelo y previniendo un posible *overfitting*.
- El balance de las clases (`class_weight`): Pesos asociados a la importancia a la que se le da reconocer una determinada clase, penalizando el error en la función de pérdida con un valor asignado en lugar de 1.

Support Vector Machines

Las Support Vector Machines (SVM) se desarrollaron inicialmente como métodos lineales de clasificación, al igual que la regresión logística. Esto significa que también buscan una frontera de decisión (*decision boundary*) que define la clase a la que pertenecen los datos usando una combinación lineal de las variables predictoras. En este caso no se busca hallar los parámetros de la distribución a modelar sino que el objetivo reside en encontrar el hiperplano (*Support Vector*) que mejor separe a los datos de entrenamiento [16]. Posteriormente este método fue extendido para buscar un hiperplano en un espacio de variables transformado (método no lineal). Esta transformación se logra reemplazando el producto interno por una función kernel. En este trabajo haremos uso de la función de base radial (RBF, por sus siglas en inglés), que es la más comúnmente usada en este método. En este algoritmo buscaremos optimizar dos hiperparámetros:

- El parámetro de penalidad (`C`): Este parámetro regula que tan bien debe el hiperplano separar las clases a costa de una distancia menor en la frontera de decisión. Es decir, un valor elevado de `C` se ajustará mejor a los datos de entrenamiento, mientras que un valor bajo será más general, a costa de errores.
- Gamma (`gamma`): Define que tan lejos alcanza la influencia de cada uno de los ejemplos de entrenamiento en la creación de la frontera de decisión.

Random Forest

A diferencia de los algoritmos anteriores, Random Forest (RF) esta basado en un método de clasificación basado en árboles de decisión. Estos métodos se caracterizan por segmentar el espacio de predicción en un número de regiones. Para entender Random Forest, resulta muy útil conocer el funcionamiento de éstos árboles.

Un árbol de decisión se construye dividiendo el espacio de variables de forma recursiva, recorriendo la lista de variables y seleccionando la variable que mejor divide las clases de acuerdo a un criterio determinado. El criterio que decidimos usar en este trabajo, por ser el más usado en tareas de clasificación, es el índice Gini (también referido como pureza):

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

donde $\hat{p}_{mk}$ representa la proporción de la clase k en la región m [16]. Uno de los principales problemas que presenta este enfoque es la alta varianza (*variance*) con respecto a los datos de entrenamiento. Si la profundidad del árbol es muy grande, aumenta el riesgo de *overfitting* en los datos que poseemos.

En el caso de Random Forest, se construyen distintos árboles de decisión con subconjuntos de variables escogidos de forma aleatoria, con el objetivo de disminuir la varianza. A la vez, al ser árboles de poca profundidad (que favorecen el *underfitting*), se genera una cantidad alta de árboles para solucionar el problema de alto sesgo (*bias*).

Esta técnica consistente en combinar distintos algoritmos para obtener un mejor predictor se la denomina *ensamble*, y en particular, Random Forest se enmarca dentro de la técnica de *bagging*. Esto significa que la predicción final se obtiene a través de un promedio de los predictores que componen el método con igual peso.

Una de las principales ventajas de Random Forest es su interpretabilidad. Esto se expresa en la posibilidad de calcular fácilmente la importancia de las variables en el modelo. La importancia de una variable (o *feature importance*) se calcula como el promedio del índice Gini en cada uno de los nodos de los árboles donde aparece, expresada proporcionalmente a la importancia máxima de todas las variables [16]. Finalmente, el algoritmos posee una serie de hiperparámetros a ser tuneados. Nosotros buscaremos optimizar los siguientes:

- Profundidad del árbol (`max_depth`): La profundidad máxima de cada árbol.
- Estimadores (`n_estimators`): La cantidad de árboles.
- Cantidad de variables por árbol (`max_features`): En cada split del árbol, se puede fijar la cantidad de variables a comparar.

Gradient Boosting y XGBoost

Gradient Boosting (GB) es otro método de ensamble, que como las otras técnicas de *boosting*, genera modelos de forma iterativa, modificando el algoritmo para intentar corregir los errores de la iteración anterior. Esta es una diferencia crucial con respecto a los algoritmos de *bagging*, como Random Forest, que genera modelos de forma paralela. La forma en que Gradient Boosting busca mejorar iterativamente es modelando el vector residual (o en otras palabras, la distancia entre la predicción y la variable a predecir) generando de esta manera un modelo aditivo que se puede representar de la siguiente manera:

$$F_m(X) = F_{m-1}(X) + \eta \cdot \Delta_m(X)$$

donde $F_{m-1}(X)$ es el modelo del paso anterior, $\Delta_m(X)$ es el modelo del vector residual y η es el *learning rate*, un hiperparámetro que busca regularizar la importancia de este factor en el modelo final para reducir el *overfitting* [17].

En este trabajo vamos a utilizar XGBoost (XGB), una implementación de Gradient Boosting con árboles muy utilizado en competencias Kaggle [15]. Este algoritmo cuenta con una gran cantidad de hiperparámetros, al ser un método de ensamble combina los hiperparámetros de los *tree boosters* con los pertenecientes al método Gradient Boosting. Decidimos limitarnos a un número relativamente pequeño de hiperparámetros buscando referencias de su uso, aunque dejamos la exploración de los restantes para un trabajo futuro. Los hiperparámetros explorados fueron:

- Peso mínimo de las hojas (`min_child_weight`): Mínima suma del peso de las instancias (hessiano) necesaria en un hoja para dejar de realizar cortes.
- Gamma (`gamma`): Mínima pérdida requerida para la creación de un nuevo corte en el árbol.
- Muestreo (`subsample`): Proporción de muestra de los ejemplos antes de la construcción del árbol.
- Cantidad de variables por árbol (`colsample_bytree`): Mismo parámetro que en Random Forest.
- Profundidad máxima (`max_depth`): Mismo parámetro que en Random Forest.

1.2.2. Pipeline de entrenamiento, validación y evaluación

Cada uno de estos métodos involucran una fase de entrenamiento, validación y evaluación. En la fase de entrenamiento los algoritmos ajustan sus datos de acuerdo a sus distintas funciones de pérdida. En nuestro trabajo también buscamos aproximarnos al mejor set de valores de hiperparámetros usando una técnica llamada *Grid-Search*. Este método consiste en evaluar distintos valores para cada parámetro, que en conjunto forman una grilla (o *grid*).

Para cada conjunto de hiperparámetros de esta grilla se entrena el modelo y finalmente se elige el conjunto de hiperparámetros que haya tenido mejor performance (bajo alguna métrica, ver sección 1.2.4). Al evaluar la performance de cada conjunto de hiperparámetros no podemos volver a usar el mismo conjunto de datos para el que fue entrenado, por lo que para evitar esto usamos otra técnica llamada *k-fold Cross Validation*. La idea es generar un corte (*split*) en el dataset de entrenamiento con el objetivo de poder evaluar los hiperparámetros en un conjunto de datos que no haya sido utilizado (conjunto de validación). Este procedimiento se realiza k veces con diferentes splits realizados al azar, con una proporción igual en cada subconjunto o *fold*. Estos *folds* son disjuntos entre sí. Una vez elegido el modelo entrenado con el mejor conjunto de hiperparámetros se evalúa en un dataset de evaluación (que no fue usado durante el entrenamiento) de acuerdo a las métricas de la sección 1.2.4.

1.2.3. Matriz de confusión

Una matriz de confusión en el caso de un problema de clasificación binario es una tabla de contingencia con cuatro posibles resultados, como podemos ver en la figura 1.6. Dado un clasificador y un conjunto de instancias del que conocemos sus clases, esta tabla informa de la cantidad de instancias correctamente clasificadas como positivas (verdaderos positivos) y la cantidad de instancias correctamente clasificadas como negativas (verdaderos negativos). También, si el valor predicho es negativo y la clase verdadera es negativa es un verdadero negativo, en caso contrario es un falso negativo [18]. Esta tabla puede expresarse en valores absolutos o relativos a la cantidad total de instancias. A partir de estos conjuntos (VP, VN, FP y FN) calculamos las métricas con las que evaluamos el modelo, descriptas a continuación.

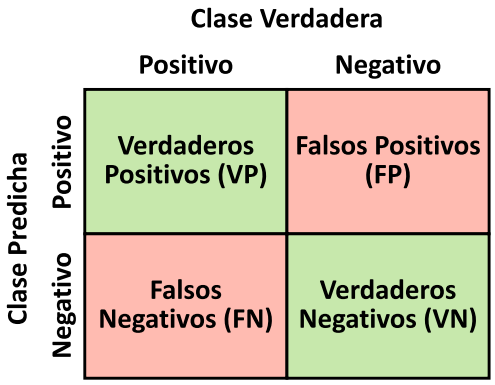


Fig. 1.6: La matriz de confusión y sus cuatro posibles casos: Verdaderos positivos, verdaderos negativos (marcados en verde), falsos negativos y falsos negativos (marcados en rojo).

1.2.4. Métricas de evaluación

Las medidas que utilizaremos para evaluar la performance del modelo son las siguientes:

- **Precisión**: La Precisión del modelo, está medida como la cantidad de positivos correctamente clasificados (VP) sobre la cantidad de instancias clasificadas como positivas: verdaderos positivos (VP) y falsos positivos (FP).

$$\frac{VP}{VP + FP}$$

- **Sensibilidad o Recall**: La Sensibilidad (o *Recall*, en inglés), corresponde a la cantidad de verdaderos positivos sobre el total de positivos: verdaderos positivos (VP) y falsos negativos (FN).

$$\frac{VP}{VP + FN}$$

- **F1-Score**: El F1-Score es el promedio armónico entre la Precisión y el Recall. Usamos el promedio armónico de manera de afectar negativamente el resultado si alguno de los valores es especialmente bajo. Posee un rango de 0 a 1.

$$F_1 = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

- **Área bajo la curva ROC (AUC)**: La curva ROC está generada por dos métricas principales, FPR y TPR:

$$TPR = \frac{VP}{VP + FN} \qquad FPR = \frac{FP}{FP + VN}$$

En un contexto de clasificación binaria, si ordenamos al score de predicción final de mayor a menor y consideramos positivos a todos aquellos a la izquierda del umbral de decisión (*threshold*), obtendremos una tasa de verdaderos positivos y de falsos positivos (TPR y FPR respectivamente). Esto puede verse en la figura 1.7.

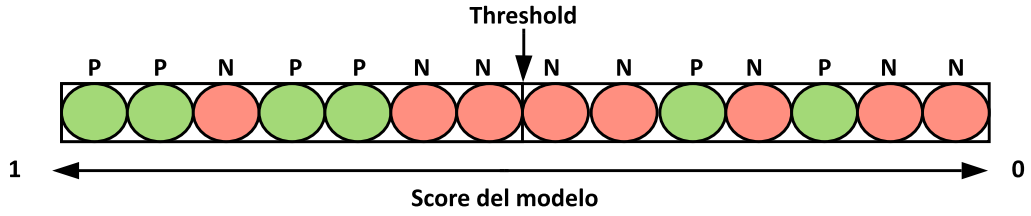


Fig. 1.7: Predicciones ordenadas de mayor a menor, donde el *threshold* asigna clasificación positiva y negativa. El color de las instancias indica el valor real. En este caso el TPR es igual a 2/3 y el FPR es igual a 3/8.

La curva característica operativa del receptor (ROC, por sus siglas en inglés) es la representación gráfica de la relación entre el FPR y el TPR al mover el umbral de decisión.

En la figura 1.8 puede verse la curva que representa el ejemplo de la figura 1.7.

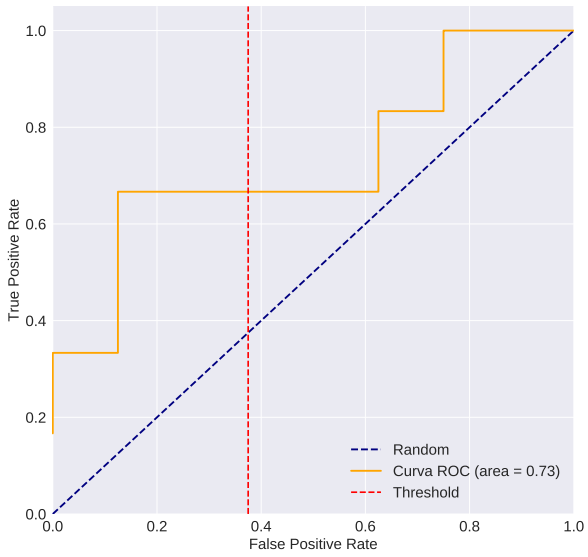


Fig. 1.8: Curva ROC de la clasificación de la figura 1.7. La línea punteada roja representa el umbral de decisión, que fijó el FPR y TPR en los valores anteriormente mencionados.

Su área (AUC, por sus siglas en inglés) equivale a la probabilidad de que el modelo clasifique una instancia positiva cualquiera "mejor" que una instancia negativa cualquiera (estadístico Mann-Whitney U [19]). Dadas dos muestras de tamaño n_1 y n_2, el estadístico Mann-Whitney U se define como:

$$U = \sum_{i=1}^{n_1} r_{1i} - \frac{n_1(n_1+1)}{2}$$

donde r_{1i} es el ranking del elemento i de la muestra n_1. La equivalencia con este estadístico viene de considerar a los verdaderos positivos y falsos positivos como muestras n_1 y n_2 respectivamente. Su distribución es aproximadamente normal para muestras grandes [20], lo que nos permite calcular su intervalo de confianza. Otro método, el test de DeLong, utiliza este mismo concepto para evaluar y comparar curvas de AUC [21].

1.3. Trabajos relacionados

A partir de mutaciones conocidas y sus propiedades asociadas, deseamos explorar un método de aprendizaje automático supervisado que nos permita generar un modelo que, ante una mutación no estudiada, pueda predecir su patogenicidad. Para recolectar datos, asociados a mutaciones conocidas, existen múltiples herramientas. En particular, para el presente trabajo vamos a explorar: VarQ, VEST y FATHMM-MKL.

VarQ

VarQ es una herramienta generada en la BIA (Plataforma Bioinformática Argentina) por Leandro Radusky [22]. Esta herramienta permite extraer datos estructurales de variantes proteicas de un sólo aminoácido (SAS, por sus siglas en inglés) tomando información de diferentes bases de datos o aplicaciones (PDB, PFam, 3DID, entre otras), permitiendo el análisis manual

de los diferentes cambios estructurales, como el tipo de actividad, el plegamiento, si pertenece a un sitio activo, o si forma parte en interfaces proteína-proteína, como se ve en la figura 1.9.

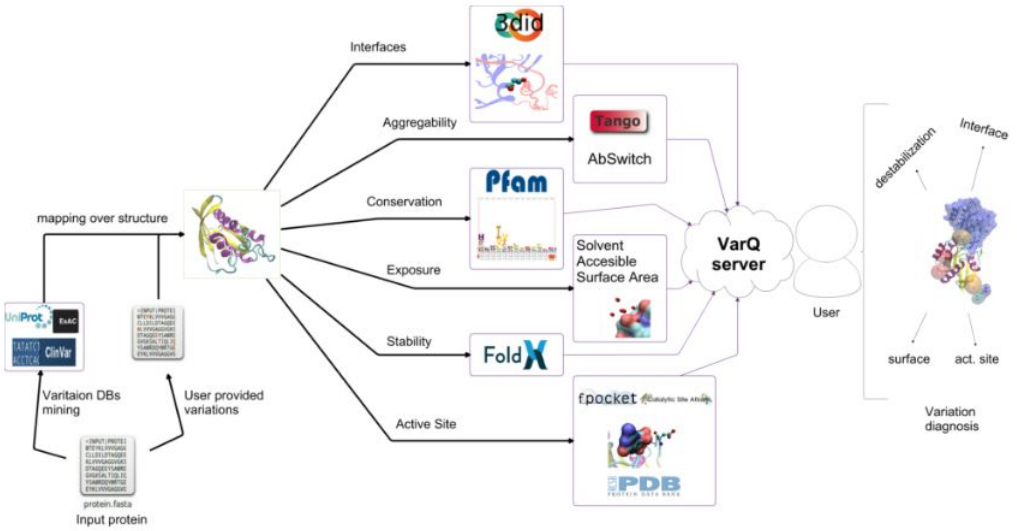


Fig. 1.9: Pipeline de extracción de datos de la herramienta VarQ. Esta figura fue extraída de la tesis de doctorado de Leandro Radusky [22].

En la primera parte de esta tesis hacemos uso del trabajo de tesis de grado de Santiago Moreno (todavía en elaboración). Uno de sus objetivos principales fue generar un dataset de variantes a nivel de proteínas (al que denominaremos dataset VarQ), y evaluar la performance de un modelo que predice el efecto de la mutación usando las variables que provee esta herramienta. En la capítulo 2 presentaremos un modelo de clasificación, variando las técnicas de aprendizaje automático presentadas, y creado en base a las variables que provee a este dataset. Posteriormente encontraremos sus limitaciones para tal fin.

VEST

VEST (*Variant Effect Scoring Tool*) [23] es un predictor de efectos funcionales de SNPs *missense* desarrollado en el Karchin Lab de la Universidad de Johns Hopkins. Con esta herramienta (basada en Random Forest) analizaron aproximadamente 80,000 variantes anotadas de HGMD (*Human Gene Mutation Database*) con variables de tipo genómicas, estructurales y físico-químicas usando la base de datos SNVBVox, desarrollado por el mismo equipo. A lo largo de nuestro trabajo integraremos muchas de las variables de esta base de datos a nuestros datasets.

FATHMM-MKL

FATHMM-MKL (*Functional Analysis through Hidden Markov Models*) [24] es también un predictor de efectos funcionales de SNPs *missense*, en regiones codificantes y no codificantes. El modelo está basado en SVM usando una combinación de múltiples kernels (*Multiple Kernel Learning, MKL*). Fue desarrollado en la Universidad de Bristol. Uno de los puntos interestantes del trabajo es un suplemento con la descripción de las variables usadas. A partir de este informe buscaremos conseguir algunas de las variables usadas que hayan tenido mayor impacto en el modelo.

1.4. Objetivos y estructura del trabajo

El objetivo del trabajo se centra en responder una serie de interrogantes generados a partir del estudio de los trabajos previos:

- El dataset de VarQ se compone esencialmente de variables de tipo estructural. ¿Es posible enriquecerlo con variables de otras dimensiones (físico-químicas, genómicas, filogenéticas)?
- ¿Cómo afectan las distintas variables a nuestros modelos de predicción de patogenicidad de los SNPs? ¿Cuáles son las más importantes para predecir una variante patogénica?
- ¿Cuáles son los mejores algoritmos de aprendizaje automático para resolver este tipo de problemas y cuáles son sus hiperparámetros?

Estos puntos serán abordados generando y estudiando modelos de aprendizaje automático sobre distintos conjuntos de variables descriptivas para los SNPs, como se describe a continuación:

- Modelo usando el dataset VarQ
- Modelo usando propiedades físico-químicas de la proteína
- Modelo usando variables genómicas
- Integración de propiedades físico-químicas y variables genómicas
- Integración de propiedades físico-químicas, variables genómicas y las variables del dataset VarQ.

2. MODELO BASADO EN EL DATASET VARQ

Comenzamos nuestro trabajo analizando el dataset construido en el trabajo de tesis de grado de Santiago Moreno (aún no finalizado). Este dataset fue construido inicialmente con las variantes originales del sitio de VarQ [25], que consistieron en aproximadamente 400 mutaciones correspondientes a 13 proteínas con 10 variables. Posteriormente en el mismo trabajo se aumentó la cantidad para llegar a las aproximadamente 18 mil variantes del dataset VarQ usando otras fuentes como Clinvar [12] y Humsavar [11]. Llamaremos a este dataset VarQ Completo.

2.1. Variables del dataset VarQ Completo

A continuación damos una descripción detallada de las variables originales encontradas en el dataset. Presentamos un extracto del trabajo de Santiago Moreno en donde se describen dichas variables:

- Variación de Energía (VARIATION_ENERGY): En VarQ, las mutaciones son modeladas con el software FoldX [26], que construye un modelo a partir de una estructura dada y luego muta residuos específicos. Luego el software predice el impacto energético de la mutación en la estabilidad de la proteína o, en caso de tratarse de un complejo, en la estabilidad del mismo.

- SASA: Es el valor correspondiente a la superficie accesible por parte del solvente, de la cadena lateral del aminoácido. Este valor permite determinar si la cadena lateral se encuentra en la superficie o en el núcleo de la estructura.

- Porcentaje de SASA: El porcentaje que representa el SASA sobre el total. Es decir el porcentaje que representa el SASA en función de la estructura de la proteína.

- B-Factor (BFACTOR): o factor de temperatura, que corresponde a un aminoácido dentro de la proteína. Una mayor temperatura, indica que el aminoácido pertenece a una zona potencialmente de mayor movilidad.

- Switchbility (SWITCHBILITY): Evalúa cuán propenso a generar un cambio de hélice alfa a hoja beta es un conjunto de aminoácidos [27].

- Aggregability (AGGREGABILITY): El software Tango [28] evalúa cuán propenso es un aminoácido a generar agregación en una proteína desde un punto de vista estructural. La agregación es el proceso por el cual proteínas mal formadas adoptan una conformación que causa su polimerización en fibrillas agregadas y organizadas. Muchas enfermedades neurodegenerativas, como por ejemplo la Amiloidosis, están asociadas con la agregación proteica.

- Conservación (CONSERVATION): Se calcula en bits, siempre y cuando la mutación pueda ser mapeada a una posición en una familia PFam asignada [29]. Cuando una posición tiene un alto valor en bits y la misma posición coincide con el aminoácido conservado en la secuencia de la proteína interpretamos que dicha posición está altamente conservada. La misma puede estar altamente conservada porque es importante estructuralmente o porque

es importante para la actividad enzimática [22]. Los residuos con alta conservación tendrán un impacto mayor sobre la función pues afectan aminoácidos de la familia.

- Sitio Activo (ACTIVE_SITE): Las posiciones de sitio activo son aquellas que se encuentran marcadas como unidas a ligandos en los archivos PDB o que pertenecen al mismo *pocket* que se encuentren conteniendo estos residuos nombrados o aquellos que pertenezcan al Catalytic Site Atlas [30].
- Interfaz 3DID (3DID): Determina si la posición sirve para una interfaz proteína-proteína según la base de datos de 3DID [31].
- Interfaz PDB (PDB): Determina si la posición sirve para una interfaz proteína-proteína según la base de datos de PDB [32].

2.2. Limpieza del dataset VarQ Completo

Para trabajar con este dataset decidimos verificar la etiqueta de cada una de las variantes, de manera de confirmar que su status de patogenicidad siguiera vigente. Para esto recurrimos a las fuentes Clinvar y Humsavar. Realizamos un primer filtrado de estas tablas quedándonos con aquellas variantes con un status confirmado: en el caso de Humsavar, aquellas que figuran con la expresión *Polymorphism* y *Disease*, mientras que en Clinvar nos quedamos con aquellas que figuraban como *Benign* y *Pathogenic*, eliminando aquellas sin un status confirmado, por ejemplo, *Risk Factor* (factor de riesgo).

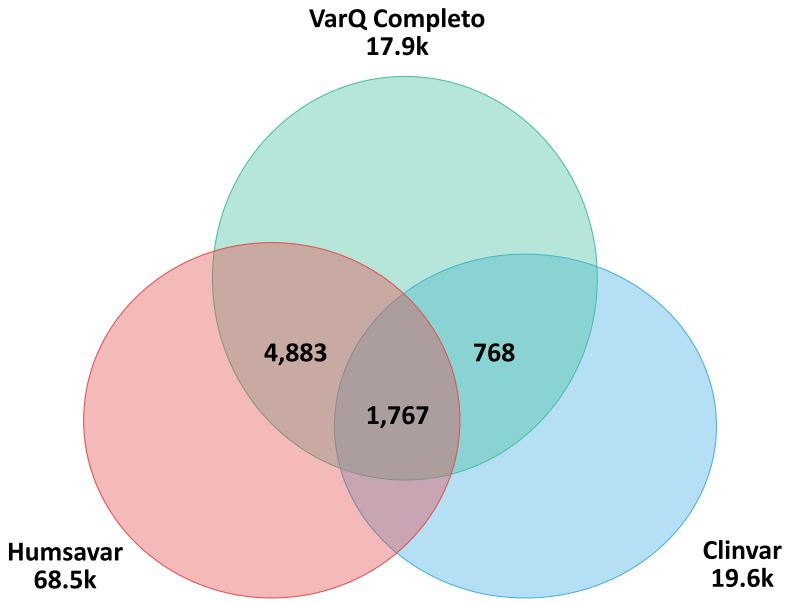


Fig. 2.1: Cantidad de mutaciones en la intersección del dataset VarQ Completo con las tablas Clinvar y Humsavar.

Así, cruzamos los datos con las tablas filtradas Humsavar y Clinvar, y encontramos un subconjunto importante de variantes que no aparecían en ninguna de las dos fuentes. Estas variantes estaban rotuladas en su gran mayoría (94 %) como benignas, por lo que hipotetizamos

que se consideraron como variantes benignas a todas aquellas a las que no se encontró un reporte. Decidimos remover estas variantes del dataset por considerar que si alguna de ellas estuviera rotulada incorrectamente introduciría ruido al modelo. Como puede observarse en la figura 2.1, de las 17,869 variantes del dataset VarQ Completo, logramos encontrar 2,535 en la tabla de Clinvar, de los cuales sólo 2,397 tenían un estado confirmado como patogénicas, y 138 como benignas. Cruzando el dataset con la tabla Humsavar encontramos una intersección de 6,650 variantes de los cuales 4,667 corresponden a patogénicas y 1,983 son benignas. Decidimos mantener la clasificación de Humsavar en la intersección de los tres conjuntos por considerarla de mayor confiabilidad dado que es un reporte único curado por expertos, a diferencia de Clinvar que es una recopilación de variantes de diversa significación clínica, y a menudo presenta conflictos de anotación por discrepancias entre evidencias reportadas. Esto nos dejó con un dataset de 7,418 variantes de las cuales 5,377 (72 %) son patogénicas y 2,041 (28 %) son benignas. Denominamos a este dataset VarQ Curado.

2.3. Descripción estadística del dataset VarQ Curado

A partir de VarQ Curado estudiaremos sus variables usando estadísticas descriptivas con el objetivo de evaluar la calidad del dataset. La idea es poder tener una noción de la dispersión de nuestros datos, sumado a la cobertura que tenemos de ellos sobre las variantes.

En la figura 2.2 podemos observar cómo la variable de sitio activo (ACTIVE_SITE) no posee datos para casi ninguna variante (aproximadamente el 95 %), mientras que la variable de conservación (CONSERVATION) no posee datos para el 63 % de las variantes. En base a estas observaciones decidimos remover la variable de sitio activo del dataset por considerarla muy poco relevante en términos de cobertura.

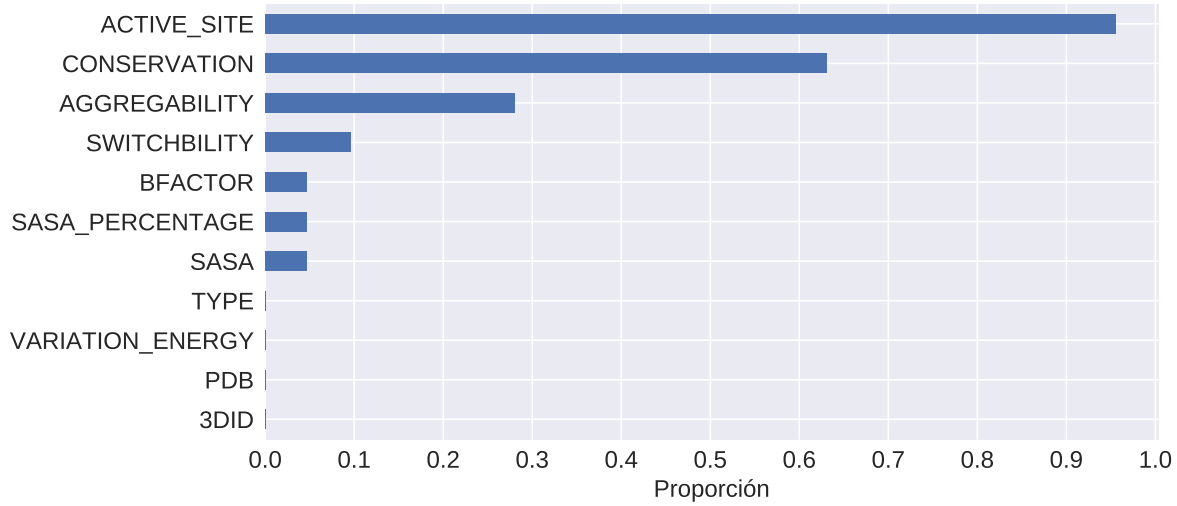


Fig. 2.2: Proporción de variantes con valor nulo por variable del dataset VarQ Curado.

Otro factor importante a considerar es cuántas variables nulas tienen cada una de las variantes del dataset. En la figura 2.3 podemos observar que existe aproximadamente un 5 % de variantes que poseen 7 variables nulas de las 10 que contienen el dataset, es decir, prácticamente no tienen ningún tipo de información, y sólo el 2 % de las variantes posee el total de las variables cubiertas. Por el otro lado, casi el 90 % de las variantes tiene a lo sumo 3 variables nulas.

Decidimos no remover ninguna fila bajo este criterio.

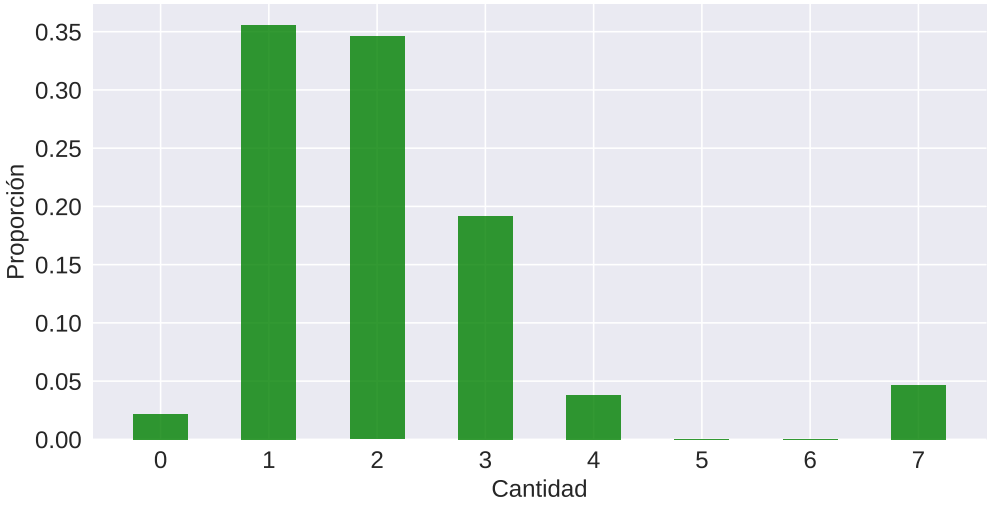


Fig. 2.3: Histograma en base a la cantidad de variables nulas por fila del dataset VarQ Curado.

También queremos conocer qué tan correlacionadas se encuentran las variables. En la figura 2.4 vemos la correlación de Spearman, que sirve para detectar relaciones monotónicas entre las variables, y así nos permite descartar variables muy similares que no aportan nueva información y ralentizan el entrenamiento del modelo. De esta forma encontramos que la variable SASA y SASA_PERCENTAGE tienen una correlación de 0.98. Mantuvimos las dos variables dado que la relativa baja cantidad de variables de este dataset no nos fuerza a removerlas, y si bien la correlación es muy alta, no descartamos a priori poder extraer información útil usando ambas.

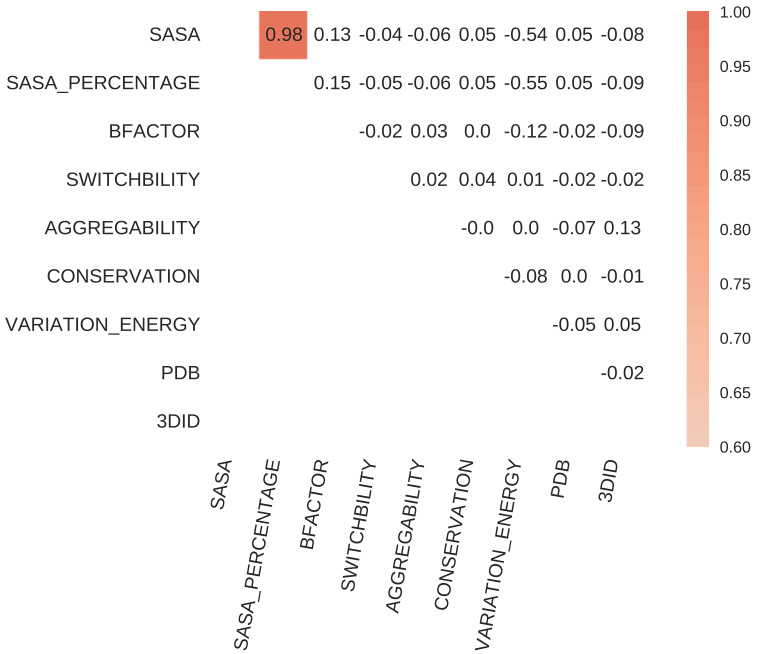


Fig. 2.4: Correlación de Spearman para las variables de VarQ Curado.

Finalmente en la tabla 2.1 podemos ver distintas métricas sobre las variables continuas del dataset, como la media de los valores (mean), el desvío estándar (std), el valor máximo (max) y los cuartiles. Estos valores nos permiten tener una idea de las distribuciones de los datos. En este caso podemos observar como la media (0.38) de SWITCH se encuentra muy alejada de su mediana (0.01), lo que nos indica que su distribución no es Normal. Por otro lado, su máximo (8.72) está también muy alejado del tercer cuartil, por lo que también posee outliers. Esto también sucede con otras variables como AGGREGABILITY y SASA.

En esta descripción sumamos el AUC univariado, es decir el área bajo la curva ROC tomando la variable ordenada como estimador de la respuesta.

Un AUC cercano a 0.5 equivale a una variable de bajo poder predictivo y un AUC mayor a 0.5 corresponde a un predictor de la variable de respuesta (en este caso una variante patogénica), mientras que un AUC menor a 0.5 corresponde a un anti-predictor, es decir, a un predictor de la clase complementaria (variantes benignas). En la columna AUC de la tabla 2.1 vemos que las variables SASA y SASA_PERCENTAGE tienen un poder anti-predictivo relativamente alto (0.34 y 0.33 respectivamente), lo que indica que las variantes que tengan un valor bajo de estas variables tienden a ser benignas. La variación de energía (ENE) tiene el mejor poder predictivo univariado entre todas las variables, lo que indica que una mutación con una alta variación posee más chances de ser patogénica.

Variable	avg	std	min	25 %	50 %	75 %	max	AUC
SASA	32.11	39.15	0.0	0.67	15.21	52.15	246.41	0.34
SASA %	0.15	0.18	0.0	0.0	0.07	0.27	0.75	0.33
BFACTOR	56.45	71.76	0.0	19.77	37.34	61.14	755.61	0.46
SWITCH	0.38	0.89	0.0	0.0	0.01	0.28	8.72	0.50
AGG	5.02	17.61	0.0	0.0	0.0	0.16	100.0	0.51
CONSERVATION	0.33	0.19	0.13	0.25	0.3	0.37	4.77	0.43
ENE	2.91	4.84	-12.64	0.26	1.51	3.89	57.21	0.68

Tab. 2.1: Descripción de variables continuas del dataset *VarQ Curado*.

Variable	top	freq. top	BACC
3DID	False	0.8	0.51
PDB	False	0.9	0.49

Tab. 2.2: Descripción de variables categóricas del dataset *VarQ Curado*.

Para el caso de las variables categóricas (PDB y 3DID), analizamos el valor con la frecuencia más alta (top) y el valor de esta frecuencia. Para este tipo de variables quisimos también cuantificar su poder predictivo individual.

Utilizar AUC en este caso no es posible dado que su cálculo sólo tiene sentido en variables continuas. Por lo tanto, decidimos calcular el *Balanced Accuracy* (BACC) [33] como medida de poder predictivo, considerando el valor de la variable como predictor de la variante.

El *Balanced Accuracy* (BACC) es igual a:

$$\frac{1}{2}(\frac{VP}{P} + \frac{VN}{N})$$

Al igual que con el AUC, un valor mucho menor a 0.5 indica poder anti-predictivo. En la tabla

2.2 podemos ver que el BACC de las variables 3DID y PDB es de 0.51 y 0.49 respectivamente, lo que indica un bajo valor predictivo.

2.4. Modelo creado a partir del dataset VarQ Curado

Una vez definido el dataset, podemos emplear técnicas de aprendizaje automático con el objetivo de generar un predictor de variantes patogénicas. Cabe destacar, que en este dataset VarQ Curado hay una sobrerrepresentación de variantes patogénicas (ver sección 2.2), es decir, que presenta un desbalance (mayor número de variantes patogénicas que benignas) que invierte las proporciones observadas en el dataset VarQ Completo. De todas formas generaremos un modelo para poder evaluar de forma preliminar la dificultad del problema.

Para esto recurrimos a diferentes algoritmos de aprendizaje automático: Support Vector Classifier (SVC) usando kernel RBF, Regresión Logística y Random Forest. La construcción del pipeline para cada uno de estos algoritmos constó de tres fases:

- **Creación del set de entrenamiento y de evaluación**: División (*split*) estratificado manteniendo las proporciones originales de la variable objetivo del dataset, 66 % para entrenamiento y 33 % para evaluación.
- **Imputación de las variables**: Se reemplazaron los valores nulos de cada variable por su mediana en caso de las continuas y por el valor más frecuente en el caso de las variables categóricas.
- **Estandarización**: Para el caso de los algoritmos paramétricos (Regresión Logística y SVC) se aplicó una estandarización robusta a outliers. Esta estandarización consiste en restar la mediana del valor y escalar los datos de acuerdo a la distancia intercuartil, como se observa en la ecuación 2.1.

$$RobustScaling(x_i) = \frac{x_i - Q_2(\mathbf{x})}{Q_3(\mathbf{x}) - Q_1(\mathbf{x})} \tag{2.1}$$

  donde x_i corresponde al valor de la variable, Q_1, Q_2 y Q_3 corresponde al primer, segundo y tercer cuartil de la variable $\mathbf{x}$ respectivamente.

Luego del preprocesamiento, para cada uno de los algoritmos se realizó una búsqueda de hiperparámetros óptimos, a partir de estos datos, con la función `GridSearchCV` de la biblioteca `scikit-learn` [14]. El objetivo de esta función es evaluar todas las combinaciones de hiperparámetros definidos en un diccionario y retornar el estimador que dio mejores resultados (de acuerdo a una métrica escogida, en este caso el área bajo la curva ROC). Esta métrica a su vez es evaluada a través de validación cruzada (*3-fold Cross Validation*). En el apéndice se encuentran los diccionarios de hiperparámetros usados en cada uno de los modelos.

2.5. Resultados

Random Forest fue el mejor modelo con un AUC de 0.74. Los parámetros óptimos de este modelo fueron una profundidad de árbol de 7, 100 estimadores y una cantidad máxima de variables por árbol de 4. Denominaremos este modelo como Modelo VarQ Curado.

La Regresión Logística y SVC obtuvieron 0.71 y 0.70 de AUC respectivamente. Estos modelos se caracterizaron ademas por tener una tendencia a clasificar a las variantes como patogénicas,

lo que se puede observar en su alto recall y menor precisión con respecto a Random Forest (ver tabla 2.3). En particular, SVC clasificó a todas las variantes del dataset de evaluación como patogénicas. También su tiempo de entrenamiento fue mucho mayor que el del resto de los algoritmos. En la tabla 2.3 presentamos la Precisión y el Recall (con respecto a las variantes patogénicas) y el tiempo de entrenamiento del modelo y de predicción de todas las variantes del dataset de evaluación.

Modelo	Precisión	Recall	AUC	F1-score	t_{fit}	t_{pred}
SVC	0.72	**1.00**	0.70	0.84	2 m 39 s	0.77s
LR	0.75	0.94	0.71	0.84	**1.17 s**	**0.01 s**
RF	**0.77**	0.93	**0.74**	0.84	9.82 s	0.11 s

Tab. 2.3: Comparación de métricas de modelos usando el dataset VarQ Curado. Las variables t_{fit} y t_{pred} corresponden al tiempo de entrenamiento y de predicción de todas las variantes. Las métricas corresponden a la identificación de variantes patogénicas.

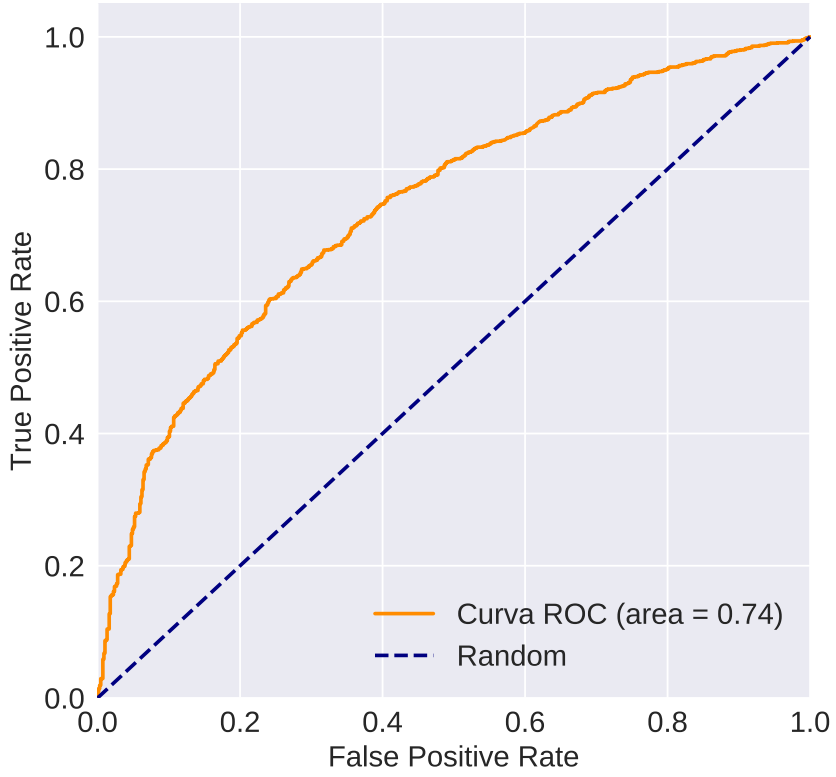
La tabla 2.4 muestra algunas métricas de interés obtenidas del modelo Random Forest para entender mejor los resultados del modelo basados en el predictor generado.

	Precisión	Recall	F1-score
Benignas	0.57	0.26	0.36
Patogénicas	0.77	0.93	0.84
Promedio	0.71	0.74	0.71

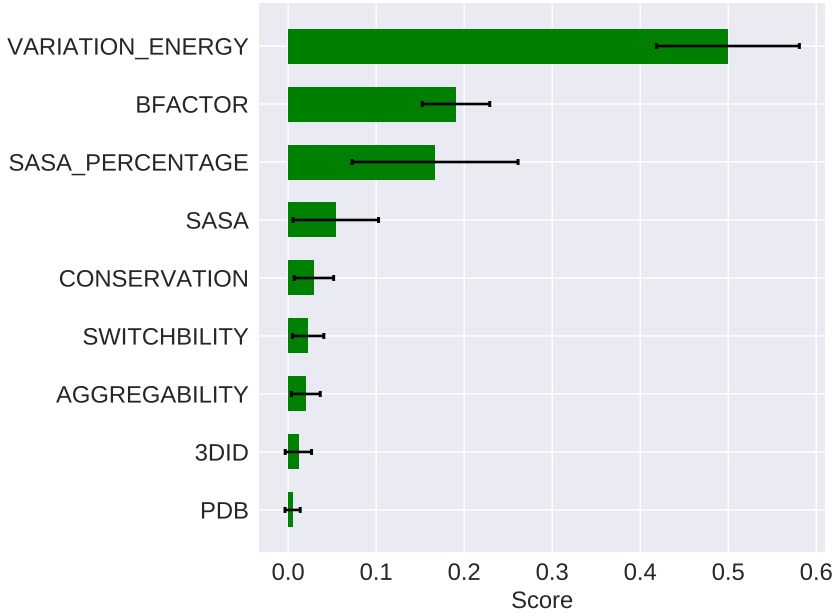
Tab. 2.4: Reporte de métricas del modelo Random Forest usando el dataset VarQ Curado. Las métricas corresponden a la identificación de variantes patogénicas, benignas y un promedio entre ambas.

La precisión del modelo indica un valor bajo (0.57) en la clase benigna, lo que significa que casi una de cada dos variantes detectadas como benignas es en efecto patogénica. El Recall con respecto a las variantes benignas es de 0.26, es decir que el modelo sólo reconoce alrededor de un cuarto de las variantes benignas como tales. Por lo tanto podemos afirmar que este modelo tiene una tendencia a clasificar las variantes como patogénicas, generando una gran cantidad de falsos positivos (o Error de tipo I). Es importante remarcar que estas métricas están generadas a partir de una función de decisión o *threshold* fijado por la versión del algoritmo usado (en el caso de la biblioteca `scikit-learn`, este *threshold* está ubicado en 0.5 de la probabilidad asignada). La curva ROC, o *Receiver Operating Characteristic* y su AUC asociada, nos permite independizarnos de un *threshold* fijo para evaluar las características del predictor (figura 2.5a).

En la figura 2.5b podemos observar la importancia de los features reportado por el algoritmo Random Forest, que ubica en primer lugar con una gran diferencia a la variable que hace referencia a la Variación de Energía (ENE), seguido por el BFACTOR (factor de temperatura) y el porcentaje de SASA (el porcentaje que representa el SASA sobre el total). Este dato concuerda parcialmente con sus valores de AUC univariado (tabla 2.1). Si bien ENE y SASA % poseen un valor relativamente alto de poder de clasificación univariada, no sucedía lo mismo con BFACTOR. En el modelo multivariado esta situación se invierte, con BFACTOR en el segundo lugar de importancia, si bien sus intervalos de confianza se superponen.



(a) Curva ROC del modelo. La línea punteada corresponde a la curva ROC de un estimador aleatorio, o *Random*, cuyo AUC es igual a 0.5.



(b) Los atributos del dataset VarQ Curado en orden de importancia. La barra de error corresponde al desvío estándar del *Feature Importance* de cada uno de los árboles del modelo.

Fig. 2.5: Curva AUC y atributos más importantes del Modelo VarQ Curado.

3. MODELO USANDO PROPIEDADES FÍSICO-QUÍMICAS DE LA PROTEÍNA

En esta sección generamos un nuevo dataset buscando fuentes complementarias de información de carácter físico-químico de las proteínas. Estas variables provienen de dos fuentes principales:

- El módulo ProtParam proveniente de la biblioteca Biopython [34]
- La base de datos SNVBox del laboratorio Karchin [35]

Para el análisis de estas variables usamos únicamente la tabla Humsavar. La tabla Humsavar (versión 2017_12) [11] está compuesta originalmente por 75,769 variantes (o "mutantes") de las cuales 39,653 son benignas (52 %), 28,855 (38 %) están asociadas a enfermedades y 7,261 (10 %) no están clasificadas. Las variantes no clasificadas fueron descartadas. Esta tabla tiene un tamaño de casi 10 veces la cantidad de variantes del dataset VarQ Curado, de aproximadamente 7400 variantes.

3.1. Extracción de variables usando Biopython

La primera fuente que utilizamos, por su relativa practicidad de uso en la extracción de un conjunto de variables físico-químicas de la proteína, fue el módulo ProtParam de la biblioteca Biopython. Esta biblioteca es un set de herramientas escritas en Python, desarrollada por un equipo internacional de desarrolladores para el área de la bioinformática, y posee una licencia de uso libre (Licencia Biopython [36]). El nombre ProtParam proviene de *Protein Parameters* (parámetros de la proteína) y está basado en la herramienta homónima del server proteómico Expasy [37]. Para poder acceder a los parámetros calculados el módulo requiere el *accession number* de la proteína (identificador único) o una subsecuencia de la misma. Las variables obtenidas son las siguientes:

- Punto isoeléctrico teórico (ISO_POINT): pH en el que la proteína (o subsecuencia) tiene carga nula.
- Aromaticidad (AROM): La frecuencia relativa de la subsecuencia Phe+Trp+Tyr (Fenilalanina, Triptófano y Tirosina).
- Índice de inestabilidad (INST): Testea la estabilidad de la subsecuencia. Cualquier valor superior a 40 indica inestabilidad, es decir una corta semivida.
- Flexibilidad (FLEX): Método de Flexibilidad implementado por Vihinen et Al [38].
- Promedio de hidrofobicidad (GRAVY): La suma de valores de hidrofobicidad de cada uno de los aminoácidos que componen la subsecuencia de la proteína.

Para poder utilizar el módulo ProtParam recurrimos a Uniprot [39] con el fin de conseguir el proteoma humano en formato FASTA [40]. El formato FASTA fue desarrollado por David Lipman y William Pearson en 1985, y originalmente fue incluido en un programa del mismo

nombre utilizado para el alineamiento múltiple de secuencias. Un archivo FASTA puede incluir diferentes secuencias, no necesariamente de aminoácidos, y cada una de estas secuencias posee una línea de descripción al comienzo que empieza con el símbolo >. Por ejemplo, así se ve la secuencia de la Ovoalbúmina, una proteína de la especie Gallus gallus (gallina), o en otras palabras, la principal proteína que encontramos en la clara de sus huevos (ver figura 3.1)

```
>P01013 GENE X PROTEIN (OVALBUMIN-RELATED)
QIKDLLVSSSTDLDTTLVLVNAIYFKGMWKTAFNAEDTREMPFHVTKQESKPVQMMCMNNSFNVATLPAE
KMKILELPFASGDLSMLVLLPDEVSDLERIEKTINFEKLTEWTNPNTMEKRRVKVYLPQMKIEEKYNLTS
VLMALGMTDLFIPSANLTGISSAESLKISQAVHGAFMELSEDGIEMAGSTGVIEDIKHSPESEQFRADHP
FLFLIKHNPTNTIVYFGRYWSP
```

Fig. 3.1: Ejemplo de código Fasta. Este código representa la proteína albúmina, donde cada letra representa un aminoácido, por ejemplo, las primeras 5 letras representan Q (glutamina), I (isoleucina), K (lisina), D (ácido aspártico) y L (leucina).

A partir del proteoma obtenido se extrajeron las secuencias correspondientes a las proteínas del dataset Humsavar, y para cada una de ellas se tomó una subsecuencia de la misma de 7 aminoácidos de largo, alrededor de la posición donde se produjo la variante (ver figura 3.2). Tomamos este largo en particular de acuerdo a su capacidad para reconocer estructuras hidrofílicas de acuerdo al trabajo de Gasteiger et al. [41], aunque dejamos para trabajos futuros la exploración de otros largos de subsecuencia.

En caso de que la variante se haya producido en los primeros o los últimos lugares, se toman aminoácidos a derecha o a izquierda según corresponda para completar el largo de la ventana.

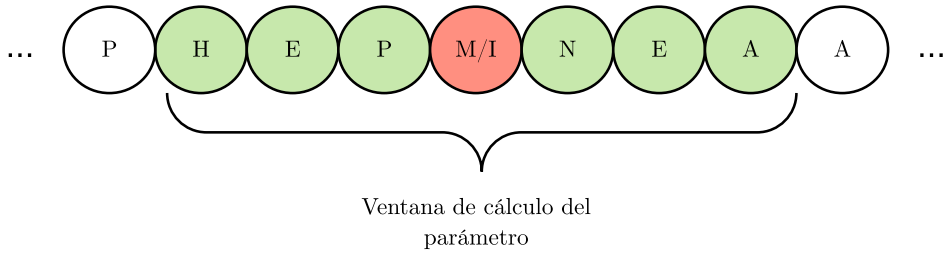


Fig. 3.2: Secuencia de aminoácidos de la proteína Arilacetamida deacetilasa (Q5VUY0). La ventana de la subsecuencia está marcada en verde, y la posición rojo es la mutación estudiada. En este caso la metionina (M) en la posición 307 fue reemplazada por una isoleucina (I).

A partir de los parámetros calculados en ProtParam, buscamos capturar la magnitud del cambio debido a la variante. Estos cambios pueden ser responsables de un efecto patogénico. Por ejemplo, si la región afectada por la variante pasa a ser hidrofílica en vez de hidrofóbica esta será más propensa a efectos adversos [42].

En este sentido, generamos dos variables que buscan reflejar la diferencia generada por la variante. Las variables son las siguientes:

- Diferencia (DIFF)

$$|x - x_{var}|$$

- Cociente de logaritmos (LOG_RATIO)

$$\frac{\log(x+1)}{\log(x_{var}+1)}$$

donde x representa al parámetro original y x_{var} es igual al parámetro de la variante.

3.2. Extracción de variables usando SNVBox

Además de la información obtenida vía ProtParam, recurrimos a una base de datos llamada SNVBox. Esta base de datos fue elaborada y es actualmente mantenida por el Karchin Lab de la Universidad Johns Hopkins. Se encuentra en su versión 3.0 y sigue actuallmente en desarrollo. SNVBox posee alrededor de 90 variables consideradas relevantes para detectar el impacto biológico de un SNV (*Single Nucleotide Variant*). Si bien existen distintos criterios para definir un SNV y su diferencia de un SNP (basados en la frecuencia del alelo menos común o MAF por sus siglas en inglés), en este trabajo los usaremos de forma indistinta. SNVBox posee datos físico-químicos de la proteína, a nivel de aminoácido y también a nivel de los sitios de la proteína donde se encuentra la variante. Otra característica destacable de esta fuente es que posee dichas variables para todos los codones del exoma humano. En el apéndice (ver sección 7.4) se encuentra el listado de variables extraídas con su descripción.

3.3. Generación del dataset Físico-Químico

Luego del proceso del extracción de variables de Biopython y SNVBox, generamos un nuevo dataset cruzando sus atributos con las variantes registradas en Humsavar. En el caso de los atributos relativos a los aminoácidos, pudimos cruzarlos usando la columna del dataset referente al identificador de la proteína, la posición de la variante, y el par de aminoácidos que se intercambiaron. Una vez agregadas todas las variantes a la tabla Humsavar, removimos todas aquellas variantes sin clasificación (etiqueta *Unclassified*). El dataset resultante (denominado dataset Físico-Químico) está compuesto por 68,508 observaciones y 50 variables, incluyendo la variable de respuesta (o tipo), de las cuales 39,653 son benignas (58 %), y 28,855 (42 %) variantes están asociadas a alguna enfermedad.

3.4. Descripción estadística del dataset Físico-Químico

Luego de la creación del dataset realizamos una exploración estadística del mismo, tal como hicimos en el dataset VarQ Curado. Calculamos la media (*mean*), el desvío estándar (*std*), el mínimo, el máximo y los cuartiles (25 %, 50 %, y 75 %). En la tabla 3.1 podemos ver que la variable GRAVY_LOG_RATIO se encuentra en una escala muy distinta a la de las demás variables, por lo que en algunos algoritmos es necesario escalar las variables (por ejemplo en la regresión logística). En el caso de la variable INST_LOG_RATIO si bien la media y la mediana son muy parecidas los valores mínimos y máximos estan muy alejados de la distancia intercuartil (tercer cuartil menos el primero), por lo que podemos afirmar que son valores atípicos o *outliers*.

También calculamos el AUC univariado para cada una de las variables continuas del dataset. Con respecto al AUC univariado en las variables de Protparam (ver tabla 3.1), notamos que

la versión DIFF de los parámetros es mejor (anti) predictor que sus versiones LOG_RATIO en todos los casos.

Variable	mean	std	min	25 %	50 %	75 %	max	AUC
AROM_DIFF	0.02	0.02	0.00	0.00	0.00	0.01	0.22	0.59
AROM_LOG_RATIO	1.97	0.33	1.00	1.94	1.94	1.94	3.66	0.53
ISO_POINT_DIFF	0.69	0.98	0.00	0.00	0.17	1.22	6.30	0.56
ISO_POINT_LOG_RATIO	2.00	0.08	1.69	1.99	2.00	2.01	2.45	0.51
GRAVY_DIFF	0.23	0.17	0.00	0.09	0.20	0.33	1.67	0.55
GRAVY_LOG_RATIO	$2x10^{12}$	$1.2x10^{14}$	$-3.3x10^{15}$	1.43	1.94	2.43	$9.6x10^{15}$	0.48
INST_DIFF	14.02	13.27	0.00	4.20	10.09	20.10	139.00	0.49
INST_LOG_RATIO	2.06	2.27	-84.77	1.96	2.01	2.09	453	0.48
FLEX_DIFF	0.01	0.01	0.00	0.00	0.01	0.01	0.05	0.54
FLEX_LOG_RATIO	2.00	0.01	1.97	1.99	2.00	2.01	2.03	0.47

Tab. 3.1: Variables extraídas de Protparam correspondientes al Dataset Físico-Químico.

Con respecto al AUC univariado de las variables extraídas de SNVBox, notamos que las mejores variables en este respecto son las aportadas por algunas de las matrices de sustitución (EX, PAM250 y BLOSUM). En este caso son buenos anti-predictores, lo que indica que un valor bajo en esta matriz aporta una mayor probabilidad de ser patogénica (ver tabla 3.2). El mejor predictor en este set de variables es el valor en la matriz de distancias GRANTHAM.

Variable	mean	std	min	25 %	50 %	75 %	max	AUC
CHARGE	0.00	0.71	-2.00	0.00	0.00	0.00	2.00	0.50
VOLUME	-0.16	1.70	-5.59	-1.40	-0.16	0.96	5.59	0.48
HYDROPHOBICITY	-0.63	6.81	-15.70	-3.10	-0.40	1.90	15.70	0.52
GRANTHAM	79.96	48.06	5.00	43.00	74.00	102.00	215.00	**0.63**
POLARITY	-0.25	2.72	-8.10	-2.20	-0.10	1.10	8.10	0.52
EX	28.99	10.95	-1.00	21.00	29.00	35.00	61.00	**0.35**
PAM250	0.16	1.68	-5.40	-1.00	0.20	1.40	5.30	**0.36**
BLOSUM	-0.58	1.65	-4.00	-2.00	-1.00	1.00	3.00	**0.35**
JM	0.80	1.24	-1.73	-0.50	1.05	1.66	3.22	0.40
VB	19.78	14.64	0.00	8.00	17.00	29.00	55.00	0.42
TRANSITION	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.47

Tab. 3.2: Variables extraídas de SNVBox relativas a sustitución de aminoácidos correspondientes al Dataset Físico-Químico.

Las variables continuas poseen una cobertura para el 100 % del dataset exceptuando algunas variables calculadas por ProtParam por alcanzar valores extremos (las cuatro variables LOG_RATIO), en los que se decidió reemplazar por nulos. En la figura 3.3 presentamos la proporción de nulos en dichas variables.

Describimos 21 variables continuas de nuestro dataset. Las 29 variables restantes son extraídas de SNVBox relativas a proteínas y son todas categóricas de tipo *Boolean* por lo que destacamos algunos datos de relevancia (ver sección 7.3 del apéndice para un listado completo).

Todas estas variables tienen una cobertura para aproximadamente el 31.5 % de las variantes del dataset, y cada una de ellas poseen valor Falso para un porcentaje superior al 90 % de las va-

riantes de la tabla, exceptuando las variables **TRANSMEM** (82 %), **REP** (87 %), **REGIONS** (70 %) y **PPI** (87 %). La descripción de estas variables se encuentra en el apéndice (ver sección 7.4).

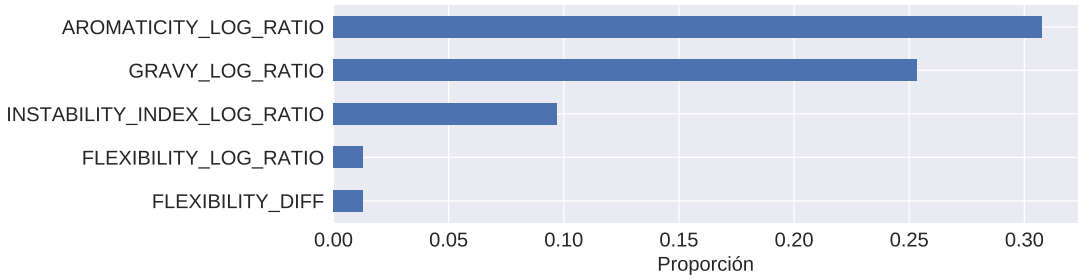


Fig. 3.3: Variables continuas con menor cobertura.

El *Balanced Accuracy* de estas variables oscilan entre el 0.45 y 0.48, lo que connota un bajo poder predictivo univariado. Por último, deseamos analizar la correlación entre las variables. Para esto usamos la correlación de Spearman dado que nos permite identificar correlaciones monotónicas. En la figura 3.4 podemos apreciar un cluster principal de variables altamente correlacionadas: Las matrices de sustitución (GRANTHAM, BLOSUM, JM, y EX).

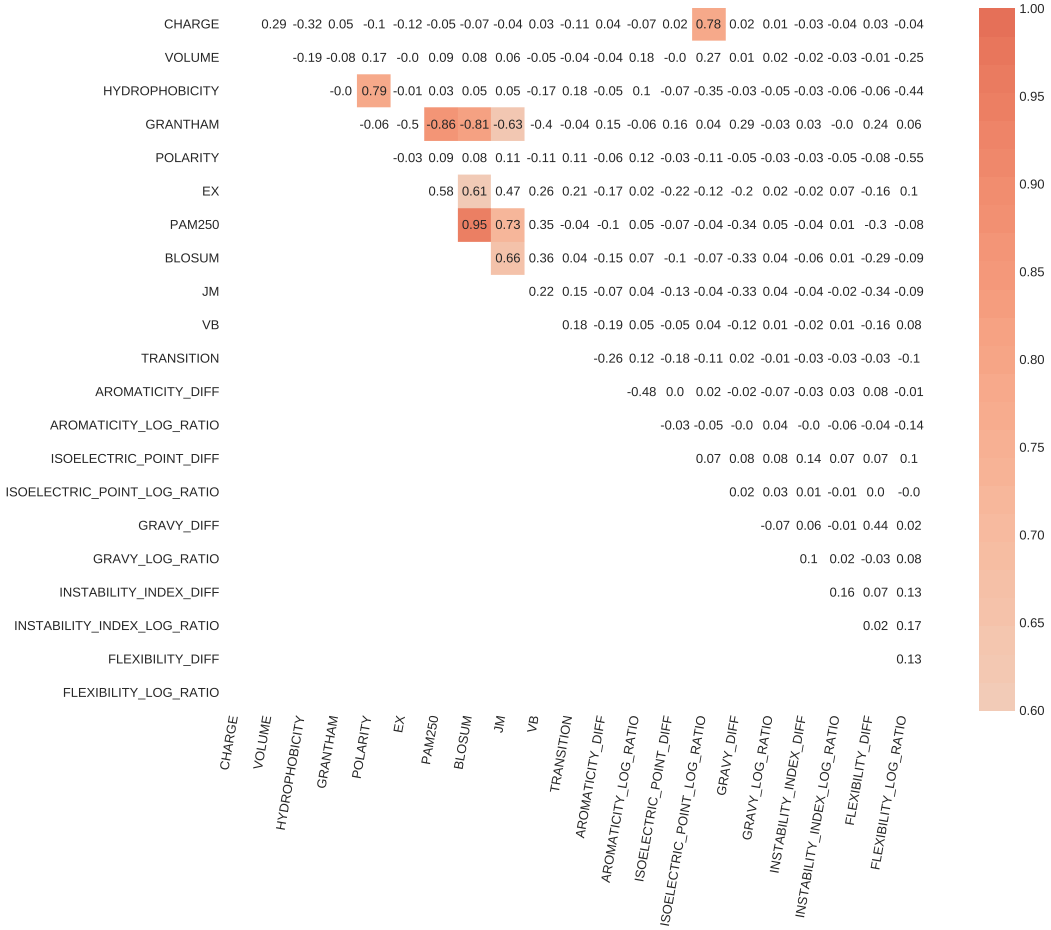


Fig. 3.4: Correlación de Spearman para las variables continuas del dataset Físico-Químico.

Por otro lado, encontramos a las variables POLARITY e HYDROPHOBICITY correlacionadas entre sí (0.79), como también a las variables CHARGE e ISOELECTRIC_POINT_LOG_RATIO (0.78). Para este análisis decidimos excluir a las variables categóricas por tratarse de variables de tipo binarias con baja cobertura, aunque de todas formas también fueron incluidas al modelo.

3.5. Generación del modelo

Con este dataset generamos un modelo basado en un predictor Random Forest. Se eligió inicialmente este tipo de predictor por ser el que obtuvo los mejores resultados en el dataset VarQ Curado con respecto a otros predictores (SVMs y Regresión Logística). Otras de las ventajas que aporta este modelo es su facilidad para explicar la importancia de las variables y su relativa baja complejidad computacional, aunque es necesario tener cuidado al respecto con las correlaciones presentes entre las variables.

Para generar el modelo volvimos a construir un *pipeline* muy similar al de la sección anterior, que consta de las siguientes etapas:

- **Imputación**: Las variables, cuyos valores eran nulos, se imputaron usando la mediana en el caso de las variables continuas, y con el valor más frecuente para las variables categóricas.
- **Escalado**: Las variables no fueron escaladas al no ser necesario en algoritmos de clasificacion basados en árboles de decisión, dado que se evalúan las variables de forma independiente.
- **Búsqueda de Hiperparámetros**: Para la búsqueda de hiperparámetros usamos *GridSearch* (búsqueda "en cuadrícula"). El diccionario de hiperparámetros de cada uno de los algoritmos se encuentran en la sección 7.3 del apéndice.

Reutilizaremos este *pipeline* en las próximas instancias en las que usemos algoritmos basados en árboles, al que denominaremos Pipeline Tree.

3.6. Resultados del modelo Físico-Químico

Como puede observarse en la figura 3.6a, a partir de este modelo se obtuvo un AUC de 0.72. Este resultado es superior a los AUCs univariados del dataset (0.35, o 0.65 si consideramos su predicción inversa), cuyos mejores valores fueron aportados por las matrices de sustitución. A su vez este modelo es superado por el modelo VarQ Curado, lo que tiene sentido si consideramos que el poder de predicción de sus variables es superior a las de este dataset, con 0.33 (o 0.67) en el caso de SASA y 0.68 en el caso de la variación de la energía (ENE).

El mejor modelo escogido durante la fase de entrenamiento posee profundidad de árbol (`max_depth`) 7, 20 % de la cantidad de variables en cada corte (`max_features`), y 100 árboles (`n_estimators`).

Las métricas observadas en la tabla 3.3 permiten dar cuenta de una precisión del 65 % con respecto a las observaciones patogénicas, es decir, el modelo está reportando un 35 % de variantes como patogénicas que no lo son (también conocido como error de tipo I), y un recall de 47 %, lo que indica que existe un 53 % de variantes patogénicas en nuestro dataset que no están siendo detectadas por nuestro modelo (error de tipo II).

	Precisión	Recall	F1-score
Benignas	0.68	0.81	0.74
Patogénicas	0.65	0.47	0.54
Promedio	0.66	0.67	0.66

Tab. 3.3: Métricas del modelo Random Forest aplicado al dataset Físico-Químico.

Si bien estos resultados son inferiores a los del modelo VarQ Curado, este modelo posee un Recall y Precisión de la clase benigna muy superior (0.81 vs. 0.26 y 0.68 vs 0.57 respectivamente), por lo que podemos afirmar que este modelo se encuentra más balanceado en la predicción de las clases, usando el *threshold* proporcionado por `scikit-learn`.

3.7. Importancia de los atributos

El algoritmo Random Forest nos permite identificar los mejores atributos en cada uno de los árboles del clasificador. En este caso, los primeros cuatro atributos refieren a matrices de sustitución (ver figura 3.6b). La quinta variable en importancia pertenece a ProtParam (AROMATICITY_DIFF). También en la figura 3.6b, se observan variables con un nivel de importancia muy similar, como es el caso de PAM250, EX, BLOSUM y GRANTHAM. Todas estas variables corresponden a matrices de sustitución. Esto último no es de extrañar ya que existe un alto nivel de correlación entre ellas (ver figura 3.4). Como mencionamos en la sección 1.2.1 (Random Forest), el score de importancia de las variables es proporcional a la importancia máxima de todas las variables. En este caso, al haber una gran cantidad de variables correlacionadas entre sí, distribuimos el score entre ellas y perdemos la importancia que pueden estar aportando otras variables al modelo.

Para solucionar este problema, acudimos a la herramienta `rfpimp` desarrollada por Terrence Parr et al. [43], que toma a su vez ideas del paper de Altmann et al. [44]. Esta herramienta permite agrupar variables y analizar su importancia realizando permutaciones aleatorias entre los ítems (SNPs), de forma de transformarlas en variables *random*. Esta permutación genera una perdida de *accuracy* en el modelo, que considera su nivel de importancia. El Accuracy se define como la cantidad de predicciones correctas dividido la cantidad de predicciones. Para generar clusters de variables observamos su nivel de correlación usando la correlación de Spearman, y agrupamos las variables que tenían una correlación mayor a 0.60. Eso nos dejo con tres clusters principales: Uno generado por HYDROPHOBICITY y POLARITY, otro por CHARGE y ISOELECTRIC_POINT_RATIO y por último otro generado por las matrices de sustitución (GRANTHAM, EX, PAM250, BLOSUM, JM y VB).

Usando `rfpimp` (ver figura 3.5) vemos más claramente como las matrices de sustitución toman un enorme rol en el desempeño del modelo, seguido por el par HYDROPHOBICITY y POLARITY. También aparecen nuevas variables en nivel de importancia, como DNA_BIND y TRANSMEM (Sitio de unión de la proteína y región transmembrana respectivamente).

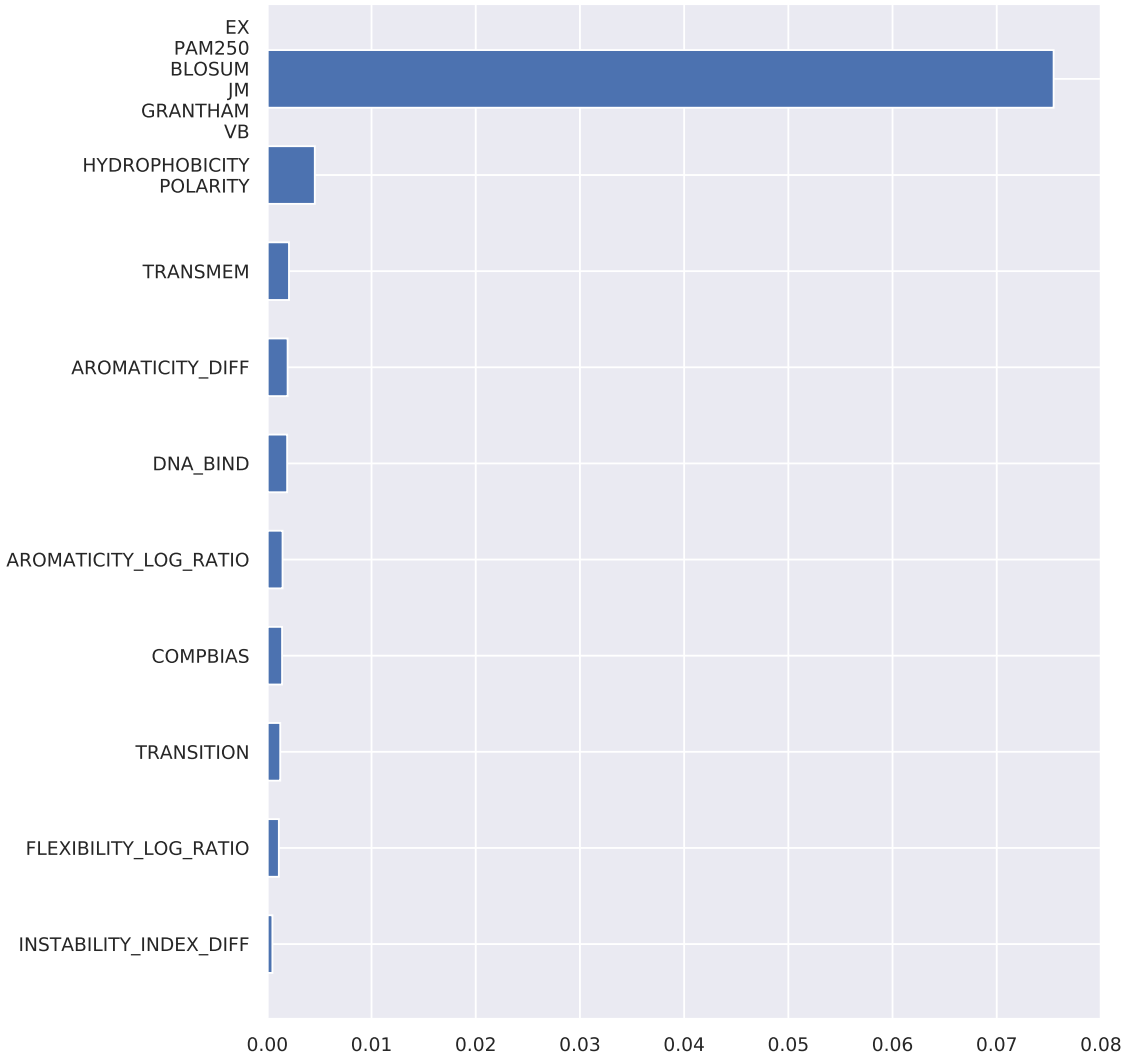
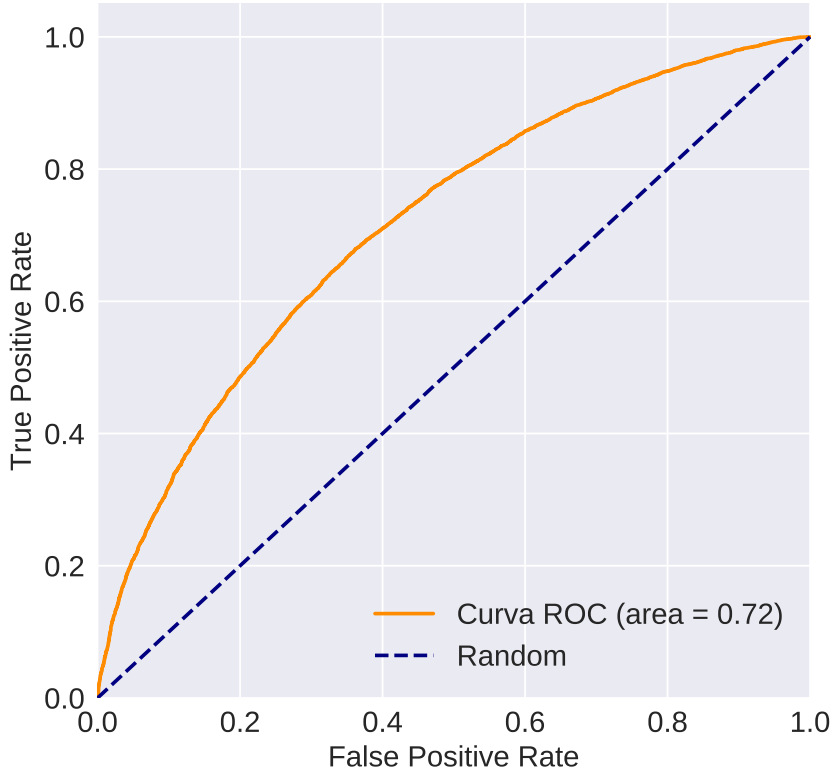
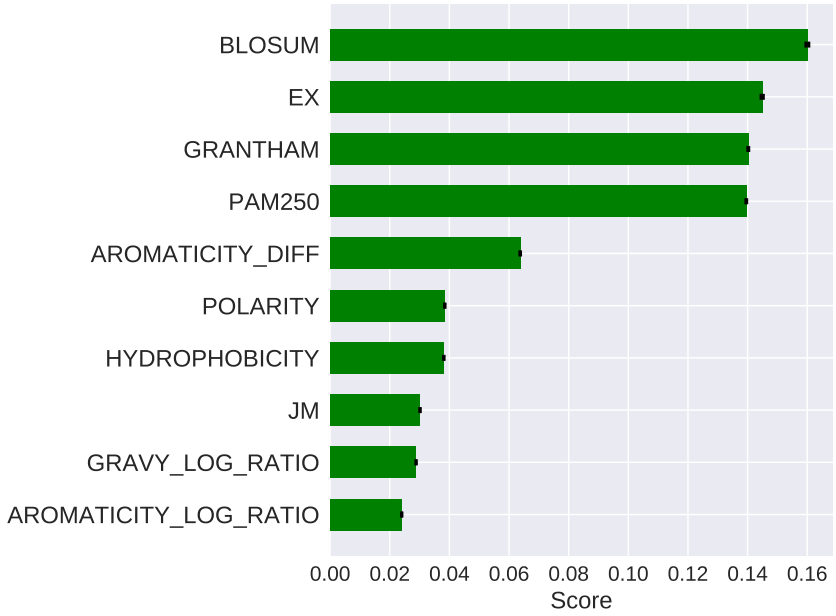


Fig. 3.5: Variación en el Accuracy al permutar clusters de variables altamente correlacionadas. Se agruparon variables con una correlación de Spearman superior a 0.60 y también se agruparon las matrices de sustitución EX, PAM250, BLOSUM, JM, GRANTHAM y VB.



(a) Curva AUC del modelo. La línea punteada corresponde a un predictor Random.



(b) Los 10 atributos más importantes del modelo según el método estándar de `scikit-learn`.

Fig. 3.6: Curva AUC y atributos más importantes del modelo Random Forest aplicado al dataset Físico-Químico. Hiperparámetros del modelo: Profundidad del árbol 7, 20 % de la cantidad de variables total en cada corte y 100 árboles.

4. MODELO USANDO VARIABLES GENÓMICAS

Otra de las preguntas que nos hicimos fue si la consideración de variables genómicas, como medidas de conservación de la posición genómica de la variante, de su entorno, del exón en que se encuentra, la cantidad de variantes observadas en el exón, etc, pueden mejorar el poder predictivo de nuestro modelo. Esta pregunta tiene lugar si consideramos que es en los genes donde se produce la mutación que finalmente da origen a la variante en la proteína. En la base de datos Humsavar existe, para la mayoría de las mutaciones, el identificador rsID o Reference SNP ID. Un identificador rsID agrupa los distintos reportes que hacen referencia a la misma posición dentro del mismo genoma de referencia (hg19/GRCh37, en nuestro caso). A partir de este identificador fue posible obtener de la base de datos dbSNP (Versión snp150) [45], datos como el cromosoma, la posición, el cambio de nucleótido de la variante y su clasificación funcional.

4.1. Variables de conservación

En la literatura encontramos que dos de las variables genómicas asociadas a la conservación eran las que daban mejores resultados de predicción (modelos FATHMM-MKL [24] y VEST [23]). La *conservación* es un concepto biológico que refiere a las secuencias conservadas, es decir, secuencias tanto genéticas como proteicas que se mantienen de forma similar o idéntica en muchas especies que poseen un ancestro evolutivo en común. La conservación puede ser cuantificada con distintos tipos de enfoques, pero en lo que nos concierne, dos de los más utilizados son 1) a través de alineamientos múltiples de secuencias, y 2) mediante el uso de árboles filogenéticos.

En particular, la composición de estas variables consiste en alineamientos múltiples de secuencias genéticas (MSA) de 46 especies de vertebrados, incluyendo Homo Sapiens y otras como Felis catus (gato doméstico), Danio renus (pez cebra) y Equus Caballus (caballo). En base a este alineamiento se usan dos medidas distintas que buscan detectar aquellas regiones en el genoma (ADN) con mayor nivel de conservación entre las distintas especies. Las medidas son las siguientes:

- **PhastCons-46-Way**: Medida de conservación basado en un modelo oculto de Markov (*phylo-HMM*). Calcula la probabilidad de que un nucleótido pertenezca a un sitio conservado (considerando el entorno) [46].
- **PhyloP-46-Way**: Mide la conservación considerando un alineamiento múltiple y un modelo de evolución neutral. Además, cada columna se mide individualmente, es decir, sin considerar sus columnas vecinas [47]

Decidimos incluir en nuestro dataset ambas medidas de conservación, usando el Table Browser de la Universidad de California en Santa Cruz (UCSC) [48].

4.2. Variables relativas a la clase funcional

También tomamos en consideración la función de la posición dentro del gen. La base de datos dbSNP define cada SNP de acuerdo a su clase funcional. Si la variación se encuentra cerca del intervalo de un transcripto, pero no en la región codificante, la clase funcional va a depender de la posición de la variación relativa a la estructura del transcripto [49]. Por otro lado, si la

variación se encuentra en una zona codificante, la clase funcional se va a definir en base a si el alelo de la variación va a resultar en una sustitución sinónima (es decir, el nuevo codón va a formar el mismo aminoácido), una sustitución *missense* (es decir, el nuevo codón va a formar un aminoácido distinto) o una sustitución *nonsense* (en donde la mutación genera un codón de terminación prematuro). En base a esta información generamos variables binarias, que indican con 1 ó 0 la existencia de un lugar con una determinada clase funcional.

Las categorías de clases funcionales que usamos en este caso fueron [50]:

- INTRON
- MISSENSE
- NEAR-GENE
- NCRNA
- CODING-SYNON
- UNTRANSLATED
- NONSENSE
- SPLICE
- STOP-LOSS

4.3. Extracción de variables usando SNVBox

Por último, usamos variables genómicas del dataset SNVBox, específicamente de la tabla EXON_FEATURES. Las variables usadas son:

- ExonConservation (CONS): Score de Conservación para el exón completo calculado a partir de una alineación filogenética a 46 vías, usando el *Genome Browser*. Si bien existe una gran similitud con las variables de conservación que ya incluimos en el dataset esta se encuentra a nivel de exón, mientras que las dos variables anteriores se encuentran a nivel de nucleótido. También decidimos incluir esta variable debido a su baja correlación con las mencionadas anteriormente (ver descripción estadística del dataset más adelante).

- ExonHapMapSnpDensity (HAPMAP_SNP_DEN): Número de SNPs (verificados en HapMap) en el exón donde ocurre la mutación dividido por la longitud del exón. Estos SNPs (también llamados tag SNPs) son los que identifican haplotipos. Los haplotipos son bloques de SNPs heredados por un único individuo.

- ExonSnpDensity (SNP_DEN): Número de SNPs en el exón donde ocurre la mutación dividido por la longitud del exón.

Estas variables se definen a nivel de exón, por lo que cada una de ellas posee dos identificadores: UID (identificador de la secuencia proteica a la que pertenece, propia de la base de datos) y EXON ID (identificador del exón dentro de cada transcripto mRNA). Estos identificadores permiten el cruce con la tabla `Transcript_Exon` dentro de la base SNVBox (ver figura 4.1). Con esta tabla podemos obtener el número del cromosoma del exón, y la posición de inicio y fin del exón dentro del cromosoma. Una vez obtenidos estos datos, pudimos extraer todos los rsID dentro de esa subsecuencia. Para aproximadamente el 33 % de los rsIDs en nuestra tabla Humsavar existe más de un transcripto que los contiene, por lo que decidimos promediar las variables (CONS, SNP_DEN y HAPMAP_SNP_DEN) de cada uno de los transcriptos.

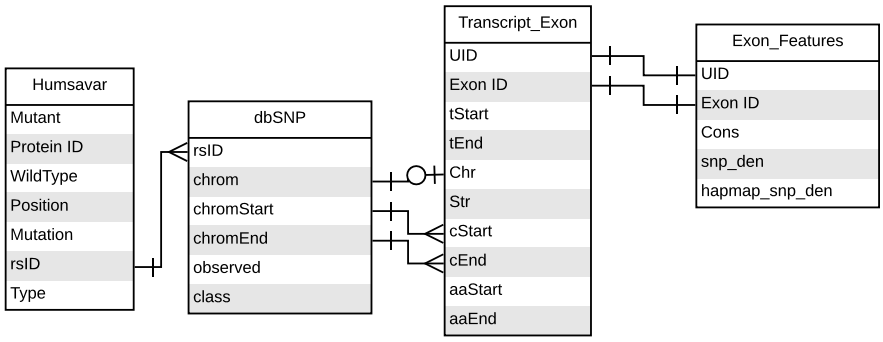


Fig. 4.1: Esquema de acceso a las variables genómicas de SNVBox. Las flechas simbolizan la cardinalidad en la relación entre las entidades: uno a muchos (la flecha con 3 puntas), uno a uno (la flecha que une UID con UID, por ejemplo) y uno a opcionalmente uno (chrom a Chr).

4.4. Construcción del dataset Genómico

Con los atributos mencionados anteriormente, pudimos cruzar la información usando la columna rsID (Reference SNP cluster ID). Esta columna identifica a un cluster de variaciones de un sólo nucleótido que pertenece a la misma posición en el genoma (o conjunto de posiciones) [51]. Filtramos las variantes de Humsavar que no poseían este identificador. Finalmente el dataset Genómico se compone de 55,382 variantes de las cuales 37,572 (68 %) son benignas y 17,807 (32 %) son patogénicas.

4.5. Descripción estadística del dataset Genómico

A continuación presentamos, como en las secciones anteriores, una descripción de las variables usadas (ver tablas 4.1 y 4.2). Para cada uno de los grupos analizamos la media (mean), el desvío estándar (std), los cuartiles (25 %, 50 %, 75 %) y los valores máximos (max) y mínimos (min). En el caso de la variable de conservación PHASTCONS46WAY encontramos que la mediana y el máximo están muy cercanos (0.99 y 1.00), por lo que podemos anticipar que los valores más interesantes (es decir, aquellos que determinen patogenicidad) se encontraran cerca del mínimo. Algo parecido sucede con la otra variable de conservación, PHYLOP46WAY, donde la mediana, el tercer cuartil y el máximo están mucho más cerca que el resto de los valores.

También calculamos el AUC univariado y el *Balanced Accuracy* para las variables categóricas, en este caso las relativas a la clase funcional. El AUC univariado de las variables de conservación (PHYLOP46WAY, PHASTCONS46WAY y CONS) es alto, lo que significa que variantes en zonas de alta conservación son buenos indicadores de patogenicidad. En el caso de las variables categóricas del dataset (las variables relativas a la clase funcional), no encontramos variables con un BACC significativo.

Variable	mean	std	min	25 %	50 %	75 %	max	AUC
PHYLOP46WAY	2.16	2.29	-8.22	0.29	1.81	4.23	6.42	0.83
PHASTCONS46WAY	0.67	0.44	0.00	0.06	0.99	1.00	1.00	0.78

Tab. 4.1: Variables de Conservación del dataset Genómico.

Variable	mean	std	min	25 %	50 %	75 %	max	AUC
CONS	0.65	0.09	0.14	0.59	0.66	0.72	0.90	0.65
SNP_DEN	0.06	0.10	0.00	0.03	0.04	0.06	1.04	0.56
HAPMAP_SNP_DEN	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.48

Tab. 4.2: Variables extraídas de SNVBox a nivel de Exón del dataset Genómico.

Con respecto a los valores nulos, tenemos una cobertura muy alta de todas las variables, con un porcentaje de nulos máximo del 2 %. (figura 4.2). Si consideramos las variantes removidas por no poseer un identificador rsID, este porcentaje aumenta.

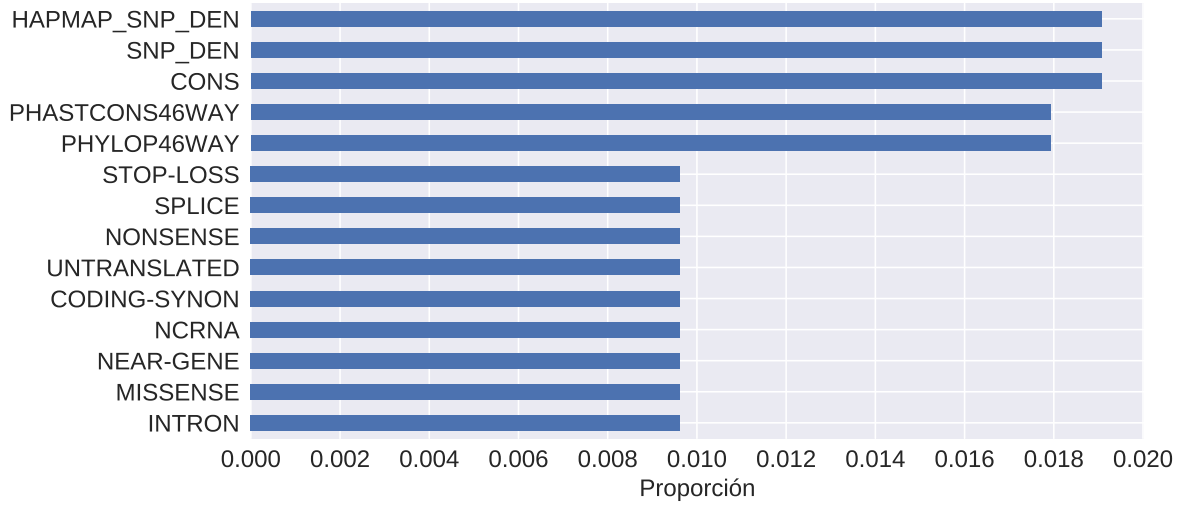


Fig. 4.2: Proporción de nulos del dataset Genómico.

Como era de esperarse, la figura 4.3 muestra una alta correlación de Spearman entre las variables de conservación PHYLOP46WAY y PHASTCONS46WAY (0.82). Por otro lado, la correlación entre HAPMAP_SNP_DEN y SNP_DEN es muy baja (0.06), pese a que su construcción es muy similar. Esto se debe a la diferencia entre la cantidad total de SNPs en un humano (alrededor de 10 millones) comparados con los que se encuentran en HapMap (aproximadamente 500,000 en total).

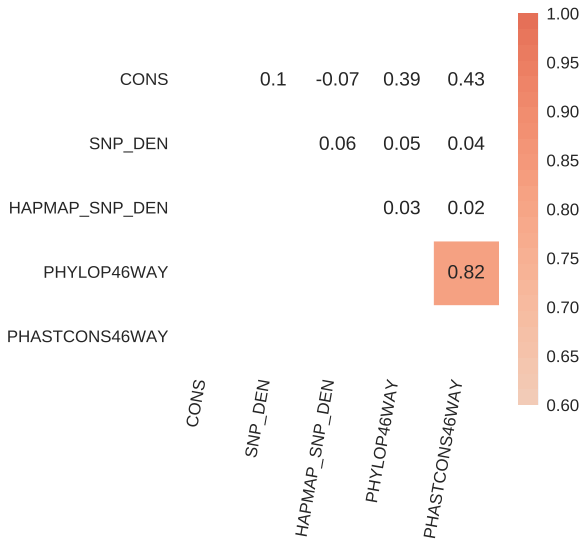


Fig. 4.3: Correlación de Spearman para las variables continuas del dataset Genómico.

4.6. Generación del modelo

Luego de realizada la exploración del dataset Genómico generamos un modelo basado en Random Forest. Volvimos a utilizar este algoritmo debido a los resultados obtenidos en el dataset VarQ Curado y para facilitar la comparación entre los datasets estudiados. Volvimos a utilizar el Pipeline Tree descripto en el capítulo anterior. Nuevamente, las variables no fueron escaladas dado que en los algoritmos que involucran árboles de decisión, como Random Forest, las variables son evaluadas una a una y por lo tanto la escala de cada una de ellas no afecta la evaluación de las demás.

4.7. Resultados del modelo Genómico

Como se puede observar en la figura 4.5a obtuvimos un AUC de 0.85. Los hiperparámetros escogidos en la fase de entrenamiento fueron una profundidad del árbol máxima (`max_depth`) de 7, una cantidad máxima de variables de 7 (`max_features`) y 100 estimadores (`n_estimators`). Este resultado es muy superior a los obtenidos en los modelos anteriores, tanto en VarQ Curado como en el dataset Físico-Químico. En la tabla 4.3 vemos los valores de Precision, Recall y F1-score para las dos clases. En este caso podemos advertir una mejora total en cada una de las métricas comparándolas con las obtenidas en base al modelo Físico-Químico. En particular destacamos la mejora en precisión de la detección de variables benignas, que pasó de un 0.68 en el modelo Físico-Químico a un 0.83 en el modelo Genómico, es decir un aproximadamente un 22 % de mejora; como así también el crecimiento en el Recall de variantes patogénicas, que saltó de un 0.47 en el modelo Físico-Químico a un 0.64, lo que representa casi un 36 % de mejora.

	Precisión	Recall	F1-score
Benignas	0.83	0.87	0.85
Patogénicas	0.69	0.64	0.67
Promedio	0.79	0.79	0.79

Tab. 4.3: Reporte de métricas del modelo Random Forest usando el dataset Genómico.

4.8. Importancia de los atributos

Analizando la importancia de las variables en el modelo, en la figura 4.5b podemos observar que las variables de conservación (PHYLOP46WAY y PHASTCONS46WAY) están en los primeros dos puestos, confirmando lo obtenido por los trabajos de investigación antes mencionados [24] [23] y su elevado AUC univariado.

El poder informativo sumado de estas variables equivale a un porcentaje superior al 80 % de la importancia total. La pregunta que nos hacemos en este caso es: ¿Las dos primeras variables de conservación están en los primeros dos lugares porque están altamente correlacionadas o aportan diferente información sobre las variables? En base al análisis de correlación de Spearman (ver figura 4.3), podemos observar que estas variables se encuentran muy correlacionadas (0.82), lo que sugiere un alto grado de redundancia. En la figura 4.4, observamos que esta proporción se mantiene al agrupar en clusters a las variables correlacionadas.

Por otro lado, esto genera un interrogante adicional: ¿Cuál es la razón por la que la variable CONSERVATION del dataset VarQ Curado no genera un rendimiento similar? Una de las principales razones que encontramos es en el nivel de cobertura de la variable. En el caso de las variables de conservación genómica, la cobertura de las mismas llega a un nivel superior al 95 %, mientras que en el caso de la conservación en el dataset VarQ Curado este número no llega al 40 %. La cobertura de estas variables posee aproximadamente la misma proporción para variables patogénicas y benignas que la proporción original en ambos datasets. Otra razón posible a considerar tiene que ver con la naturaleza del cálculo de conservación de VarQ via PFAM, que contiene una heterogeneidad muy grande de especies (alrededor de 16,000 familias), y se basa en secuencias de proteínas, mientras que PHYLOP46WAY y PHASTCONS46WAY se basa en secuencias genómicas.

En resumen, hemos encontrado un modelo que supera en AUC ampliamente a los modelos anteriores. Sin embargo, no hay un salto significativo entre el AUC univariado (0.83 en PHYLOP46WAY) y el resultado final al combinar el resto de las variables y el algoritmo Random Forest. En el próximo capítulo evaluaremos la combinación de los datasets usando Humsavar y la posible mejora de un nuevo algoritmo, XGBoost.

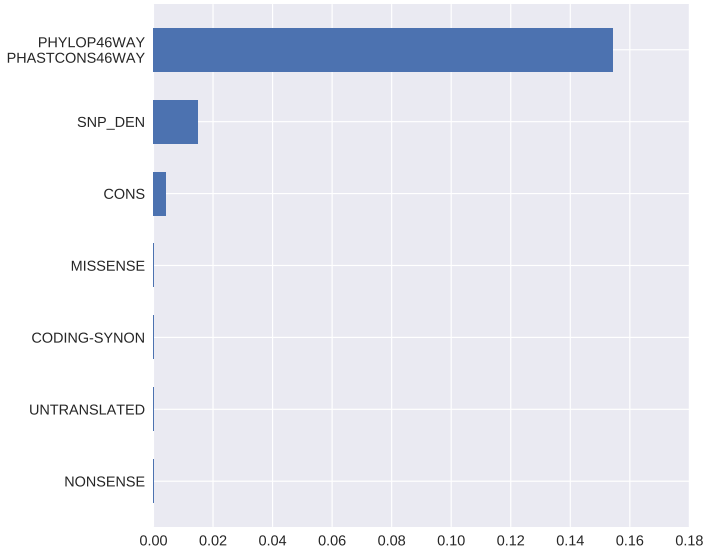
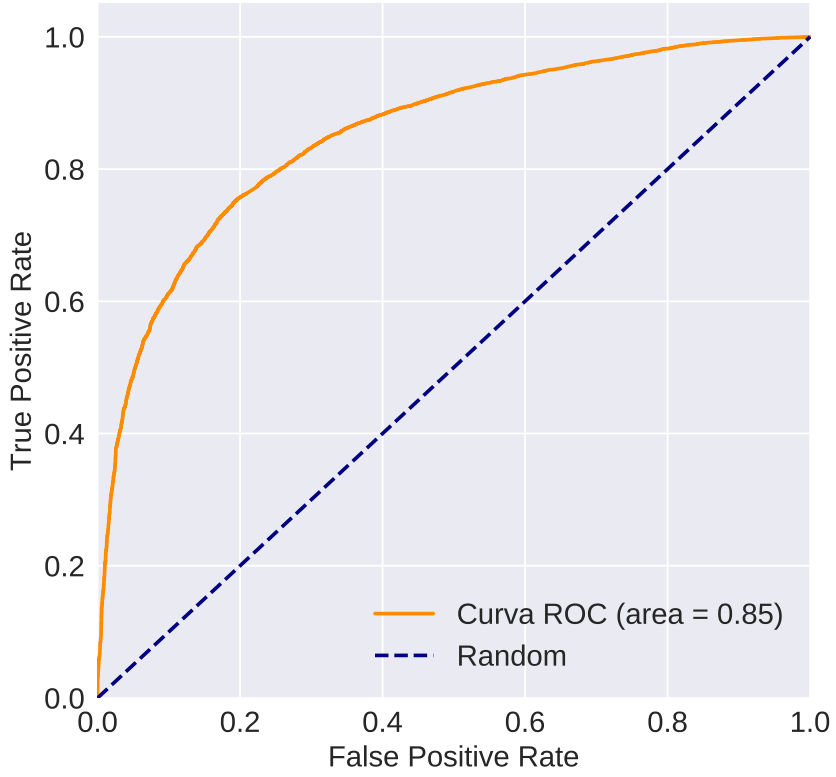
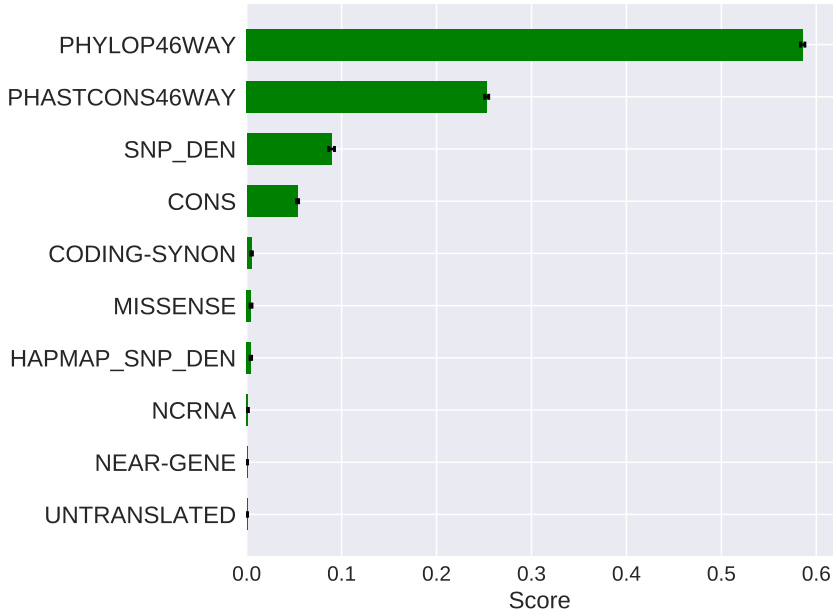


Fig. 4.4: Variación en el Accuracy al permutar clusters de variables correlacionadas (con un valor de correlación de Spearman superior a 0.60) del dataset Genómico.



(a) Curva AUC del modelo. La línea punteada corresponde a un predictor Random.



(b) Los 10 atributos más importantes del modelo.

Fig. 4.5: Curva AUC y atributos más importantes del modelo Random Forest aplicado al dataset Genómico. Hiperparámetros del modelo: Profundidad del árbol 7, 7 variables en cada corte y 100 árboles.

5. INTEGRANDO EL DATASET FÍSICO-QUÍMICO Y EL GENÓMICO

En este capítulo unimos los dos conjuntos de variables para evaluar si la integración de ambos datasets representan una mejora frente a los resultados de los modelos Genómico y Físico-Químico por separado. A este nuevo dataset lo denominamos dataset Integral. Las variantes usadas fueron nuevamente las encontradas en la tabla Humsavar.

5.1. Creación del dataset Integral

El dataset Integral posee 68,508 variantes. Este número equivale a la cantidad de variantes del dataset Físico-Químico, y esto se debe a que conservamos todas sus variantes sumando variables del Dataset Genómico (ver figura 5.1). Esto da un total de 64 variables, que son las variables sumadas de los datasets Genómico (14), Físico-Químico (49) y la variable de respuesta.

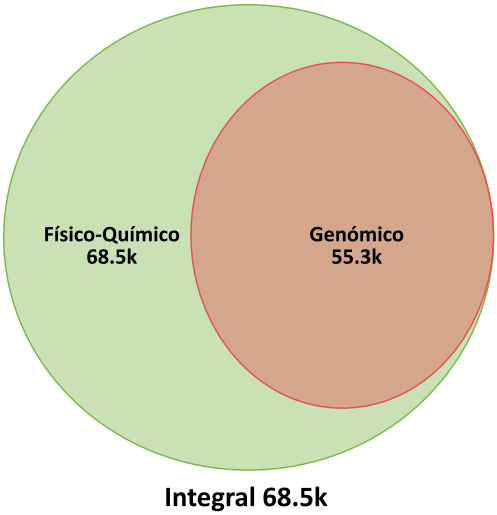


Fig. 5.1: Intersección entre los datasets Genómico y Físico-Químico.

Del total de las variantes, 39,653 (58 %) son benignas y 28,855 (42 %) son patogénicas. Con respecto a los nulos, mantenemos la misma cobertura de las variables físico-químicas del dataset homónimo (aproximadamente 35 % para las variables categóricas y una cobertura mayor del 60 % para las continuas), mientras que para las variables genómicas tenemos alrededor de un 20 % de variantes que no poseen cobertura al no tener un identificador rsID.

5.2. Generación del modelo

5.2.1. Random Forest

Para este modelo preliminar utilizamos nuevamente el Pipeline Tree, repitiendo las mismas fases de imputación y entrenamiento usadas previamente en los modelos Genómico y Físico-Químico. Esto significa que en la fase de imputación usamos la mediana para los valores nulos de las variables continuas y el valor más frecuente en las variables categóricas. El entrenamiento

consistió en una búsqueda de hiperparámetros óptimos usando *GridSearch* en un diccionario de posibles parámetros (ver sección 7.3 del apéndice), evaluados en una triple validación cruzada basándonos en el AUC como métrica a optimizar. El dataset de entrenamiento posee 45,900 variantes, aproximadamente dos tercios del dataset completo. Una vez entrenado el modelo usando este *pipeline*, se evaluaron las variantes del dataset de testeo, es decir, el tercio restante del dataset completo.

5.2.2. XGBoost

Posteriormente introducimos un método de boosting, XGBoost, nuevamente usando el Pipeline Tree. Los hiperparámetros de este método fueron elegidos usando una búsqueda randomizada (*Randomized Grid-Search*). Este método de optimización fue elegido dado que las combinaciones que se evalúan en el método Grid Search son demasiadas (ver sección 7.3 del apéndice para diccionario completo de hiperparámetros explorados). La búsqueda randomizada evalúa las mismas alternativas pero eligiendo combinaciones de forma aleatoria, sin probar todas las combinaciones. También es posible evaluar valores tomados de acuerdo a una distribución específica. Decidimos dejar este análisis para trabajos futuros.

5.3. Resultados del modelo Integral

5.3.1. Modelo usando Random Forest

El modelo Random Forest obtuvo un AUC de 0.88 (ver figura 5.3a). Esto representa una mejora con respecto al modelo Genómico (0.85) y al modelo Físico-Químico (0.72). Los hiperparámetros escogidos fueron: profundidad del árbol 7 (`max_depth`), una cantidad máxima de variables del 20 % de las variables predictoras (`max_features`) y 100 árboles (`n_estimators`). Si comparamos las métricas obtenidas en el modelo Genómico con las de este modelo (ver tabla 5.1), se puede apreciar un nuevo salto en el Recall de las variables patogénicas, que pasa de un 0.64 en el modelo Genómico al 0.80, lo que representa un 25 % de mejora. También mejora la precisión con respecto a la detección de esta clase, que pasa de 0.69 a 0.76. La única métrica que decae es el Recall de la clase Benigna, que pasa de 0.87 a 0.82, lo que representa una leve caída del F1-score.

Clase	Precisión	Recall	F1-score
Benignas	0.85	0.82	0.83
Patogénicas	0.76	0.80	0.78
Promedio	0.81	0.81	0.81

Tab. 5.1: Reporte de métricas del modelo Random Forest usando el dataset Integral.

5.3.2. Modelo usando XGBoost

El modelo XGBoost superó la performance de Random Forest alcanzando un AUC de 0.90. Las otras métricas (Precisión, Recall, F1-score) también fueron levemente superiores en todos los casos, tanto para variables benignas como patogénicas (ver tabla 5.2). Los hiperparámetros obtenidos en el *Randomized Grid-Search* fueron:

- `min_child_weight`: 5

- `gamma`: 5
- `subsample`: 0.8
- `colsample_bytree`: 0.8
- `max_depth`: 5

Utilizando el test de DeLong para comparar los AUCs del modelo RF y XGB obtuvimos un p-valor igual a 2e−16. Esto indica que las diferencias observadas en los valores de AUC pueden deberse a fluctuaciones aleatorias con probabilidad menor a 2e−16.

Clase	Precisión	Recall	F1-score
Benignas	0.86	0.83	0.84
Patogénicas	0.78	0.82	0.80
Promedio	0.83	0.82	0.82

Tab. 5.2: Reporte de métricas del modelo XGB usando el dataset Integral.

5.3.3. Comparación entre los modelos

En la tabla 5.3 comparamos la Precisión, el Recall y el AUC, los tiempos de entrenamiento y de evaluación. Los tiempos de entrenamiento incluyen todas las variantes del set de entrenamiento, usando 3 folds en la etapa de validación y la búsqueda de hiperparámetros. El tiempo de evaluación equivale al tiempo de todas las variables del set de evaluación. Las métricas están basadas en las variantes patogénicas como variantes positivas. El modelo XGB supera al modelo RF en casi todas las métricas exceptuando al tiempo de entrenamiento, que incluso usando el método de busca de hiperparámetros randomizado resultó ser mucho más lento que el modelo RF, si bien es posible reducir aún más el espacio de búsqueda. Esto se debe mayormente a que a diferencia de RF que genera estimadores de forma paralela, XGB es iterativo, lo que ralentiza el proceso.

Modelo	Precisión	Recall	AUC	F1-score	t_{fit}	t_{pred}
RF	0.76	0.80	0.88	0.78	2m 2 s	0.3 s
XGB	0.78	0.82	0.90	0.80	12m 47s	1.14 s

Tab. 5.3: Comparación de métricas de modelos usando el dataset Integral. Las variables t_{fit} y t_{pred} corresponden al tiempo de entrenamiento y de predicción de todas las variantes

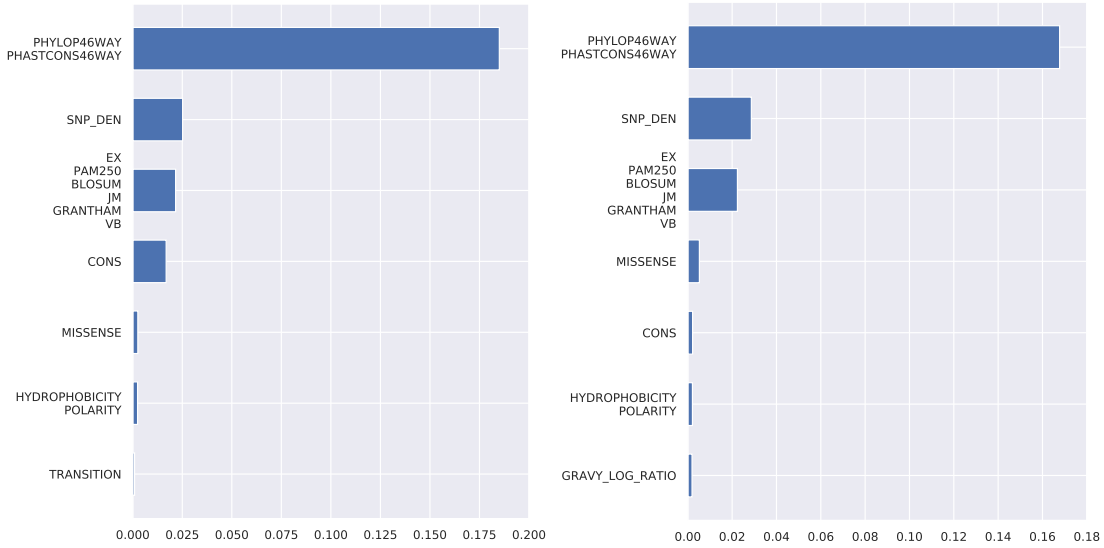
5.4. Importancia de las variables

Al combinar los dos datasets Genómico y Físico-Químico volvimos a evaluar la importancia de las variables en los modelos, dado que las nuevas interacciones entre ellas pueden haber modificado los resultados anteriores.

Si analizamos solamente la importancia usando el método de `scikit-learn` para el modelo RF (ver figura 5.3b), nuevamente encontramos en primer lugar a las variables de conservación genómicas (PHYLOP46WAY y PHASTCONS46WAY), resultado esperable dado su nivel de importancia en el dataset Genómico y su nivel de AUC conseguido. También encontramos en un segundo escalón a las variables de conservación a nivel de exones, y a una variable que considera

el número de SNPs en el exón donde ocurre la mutación. Luego encontramos al grupo de matrices de sustitución de aminoácidos (EX, GRANTHAM, BLOSUM y PAM250), que también aparecieron en los primeros lugares en el modelo Físico-Químico. Por último encontramos una variable relativa a la clase funcional a nivel genómico (MISSENSE) y otra relativa al cambio de polaridad del aminoácido donde ocurre la variante (POLARITY) y a la hidrofobicidad del aminoácido (HYDROPHOBICITY), por lo que encontramos en nuestro ranking una lista de variables transversal a los dos datasets usados.

En las figuras 5.2a y 5.2b unimos las variables de alta correlación en clusters y comparamos su impacto en la precisión del modelo usando la herramienta `rfpimp`. Notamos que la importancia de CONS se ve disminuida en el modelo XGB con respecto al modelo RF.

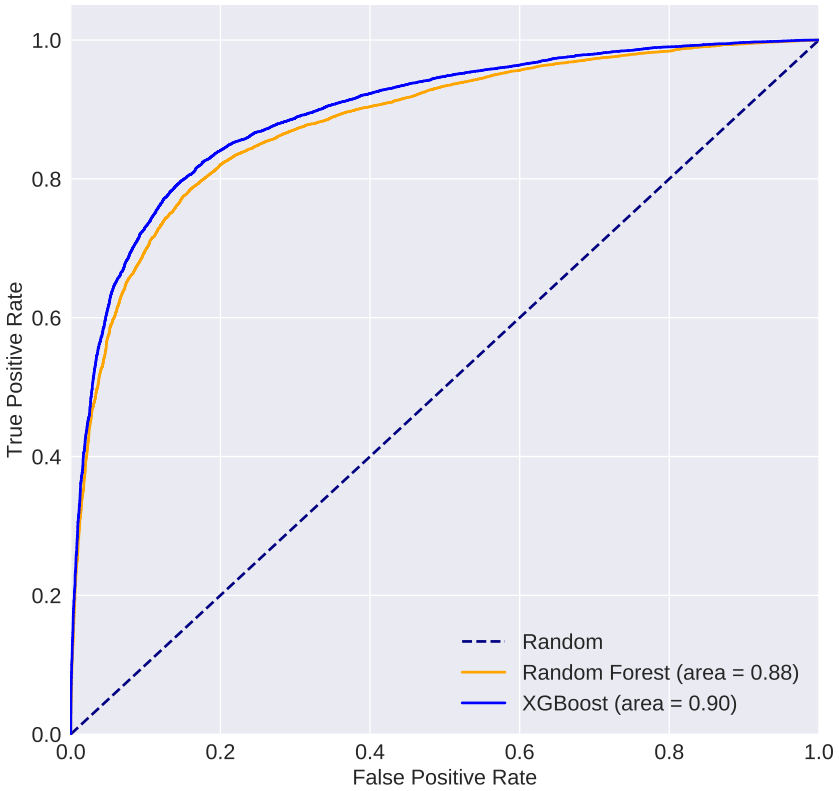


(a) Resultados del modelo usando RF. **(b)** Resultados del modelo usando XGB.

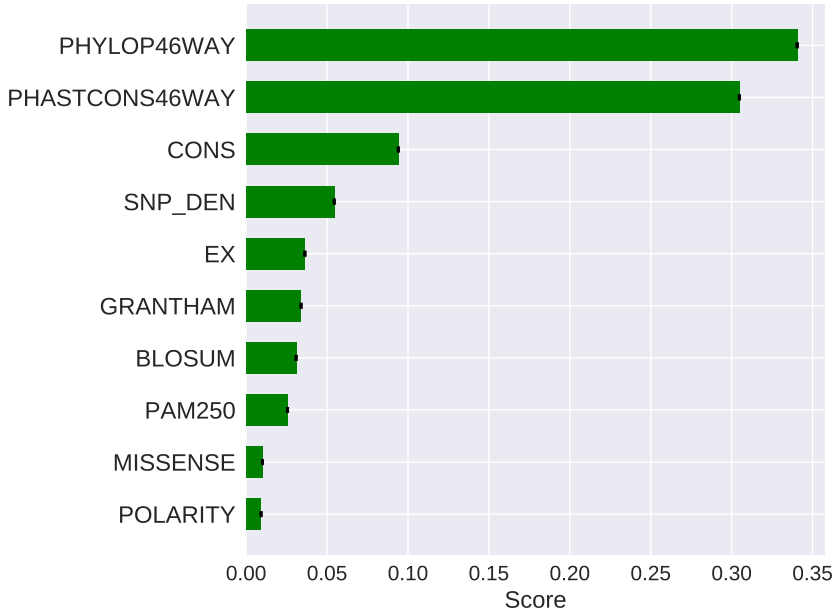
Fig. 5.2: Importancia de variables altamente correlacionadas del dataset Integral (basados en correlación de Spearman) usando permutación.

5.5. Conclusión del capítulo

Como conclusión de este capítulo, resaltamos la mejora aportada tanto por el dataset Físico-Químico como por el algoritmo XGBoost. Por un lado, mantuvimos constante el algoritmo usado en la sección anterior, Random Forest, al que incluimos nuevas variables, y eso significó un salto en el AUC de 0.85 a 0.88. En un segundo paso, modificamos el algoritmo manteniendo las mismas variables, consiguiendo un AUC de 0.90. Consideramos que igualmente hay espacio para mejorar aún más, y evaluaremos la incorporación de variables estructurales en el próximo capítulo siguiendo el mismo esquema.



(a) Comparación de curvas AUC entre algoritmos Random Forest y XGBoost. La línea punteada corresponde a un predictor Random.



(b) Los 10 atributos más importantes del dataset Integral aportados por el algoritmo Random Forest.

Fig. 5.3: Curva AUC y atributos más importantes de los modelos RF y XGBoost usando el dataset Integral.

6. INTEGRANDO LAS NUEVAS VARIABLES AL DATASET VARQ CURADO

En esta sección buscamos cuantificar en qué medida el esfuerzo realizado a lo largo de esta tesis mejora e impacta sobre nuestro set de datos original (VarQ). Para ello, integraremos al set VarQ Curado, que dispone de 9 features estructurales y 7,418 variantes, los features físico-químicos y genómicos obtenidos a lo largo de esta tesis.

6.1. Creación del dataset Integral+VarQ Curado

Para generar este dataset cruzamos las variantes de ambos datasets, haciendo un *right-outer-join* (ver figura 6.1). Es decir que nos quedamos con las variantes de VarQ Curado a las que sumamos las variables del dataset Integral para aquellas variantes en la intersección de los dos conjuntos. El dataset resultante posee 73 variables, que corresponden a las 63 variables del dataset Integral sumado a las 9 variables del dataset VarQ Curado y la variable de respuesta. Este dataset posee 7,418 variantes de las cuales 5,377 (72 %) son patogénicas y 2,041 (28 %) son benignas. Las 774 variantes que no poseen variables del dataset Integral se mantuvieron en este nuevo dataset. Es por eso que en este caso la proporción de nulos para las variables genómicas es mayor que en el caso Integral, por un lado porque no se recomputaron esas variables para las variantes que no están en la intersección (774), si no que además dentro de la intersección con el dataset Integral, la cantidad de variantes sin cobertura genómica es de aproximadamente el 36 %, mientras que en el dataset Integral este valor llegaba al 20 %. Dejamos como trabajo futuro la generación de variables genómicas para las variantes de VarQ Curado no presentes en el dataset Integral.

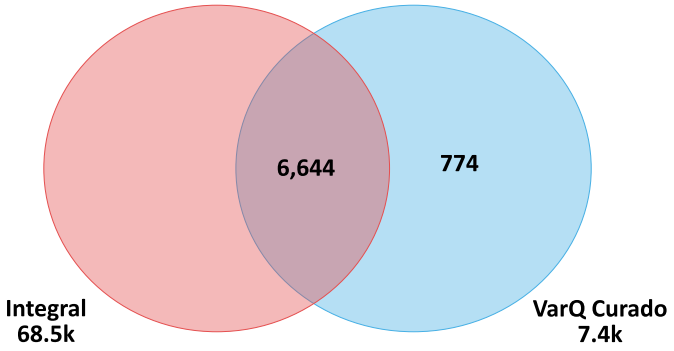


Fig. 6.1: Intersección entre los datasets Integral y VarQ Curado .

6.2. Generación del modelo

Como en los capítulos anteriores, se volvió a utilizar el Pipeline Tree para los modelos XGBoost y Random Forest. El dataset de entrenamiento posee 4,970 variantes (66 %), y el tercio restante se destinó al dataset de test. Estas variantes fueron elegidas al azar, con una

semilla pseudoaleatoria para poder replicar el experimento. Este procedimiento se repitió en todos los modelos.

6.3. Resultados

El dataset de test arrojó un AUC de 0.86 para Random Forest y 0.88 con XGBoost (ver figura 6.3a). El test de DeLong [21] arrojó un p-valor menor a 2e−7, lo que nos permite aseverar que el modelo XGBoost es superior a Random Forest. Este resultado representa una mejora sensible con respecto al modelo realizado con el dataset VarQ Curado (0.74), sin superar lo obtenido por el dataset Integral (0.90 con XGBoost). Nuestra hipótesis es que esto se debe a la menor cobertura de las variables de conservación genómica. Los hiperparámetros de los modelos fueron, en el caso de Random Forest: Profundidad del árbol 7, 100 estimadores y Cantidad de variables por árbol $0.2*n$ con n la cantidad total de variables. En el caso de XGBoost, los hiperparámetros elegidos fueron:

- `min_child_weight`: 5
- `gamma`: 1.5
- `subsample`: 1
- `colsample_bytree`: 0.6
- `max_depth`: 5

Considerando a la clase patogénica como clase positiva, vemos que XGBoost supera a RF en Precisión, AUC y F1-score, pero no en Recall (ver tabla 6.1). Sin embargo, si tomamos a las clase benigna como positiva, el modelo XGBoost supera a RF (0.59 vs 0.53). Si bien sigue siendo un número bajo, es posible modificar el *threshold* en la función de decisión en ambos modelos para obtener un Recall más alto sacrificando precisión.

Modelo	Precisión	Recall	AUC	F1-score	t_{fit}	t_{pred}
RF	0.84	0.95	0.87	0.89	15.4 s	0.07 s
XGBoost	0.86	0.94	0.88	0.90	1m 20 s	0.1 s

Tab. 6.1: Comparación de métricas de modelos usando el dataset Integral+VarQ Curado. Las variables t_{fit} y t_{pred} corresponden al tiempo de entrenamiento y de predicción de todas las variantes.

	Precision	Recall	F1-score
Benignas	0.81	0.53	0.64
Patogénicas	0.84	0.95	0.89
Promedio	0.83	0.83	0.82

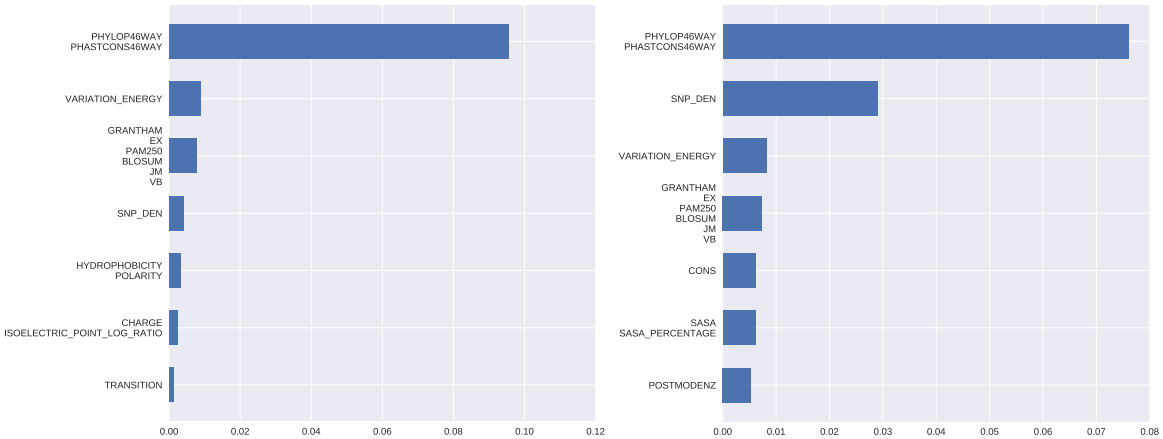
Tab. 6.2: Reporte de métricas del modelo Random Forest usando el dataset Integral+VarQ Curado.

	Precision	Recall	F1-score
Benignas	0.80	0.59	0.68
Patogénicas	0.86	0.94	0.90
Promedio	0.84	0.85	0.84

Tab. 6.3: Reporte de métricas del modelo XGB usando el dataset Integral+VarQ Curado.

6.4. Importancia de las variables

La información proporcionada por `scikit-learn` acerca de la importancia de las variables en el modelo Random Forest están presentadas en la figura 6.3b. En este ranking de las 10 variables más relevantes encontramos otra vez en primer lugar con amplia ventaja a las variables de conservación genómica, aunque también se mantiene la variación de energía y al porcentaje de SASA, que son variables pertenecientes al dataset VarQ Curado y que habían aparecido en el ranking de dicho modelo. Si consideramos la variaciones en la precisión de los modelos Random Forest y XGBoost calculados por el módulo `rfpimp` (ver figuras 6.2a y 6.2b), encontramos nuevamente a SNP_DEN como una variable relevante en el modelo XGBoost, mientras que en el modelo RF aparece en el cuarto puesto con escasa diferencia de las variables con menor relevancia. VARIATION_ENERGY y las matrices (GRANTHAM, EX, PAM250, etc.) aparecen en ambos modelos como relevantes.

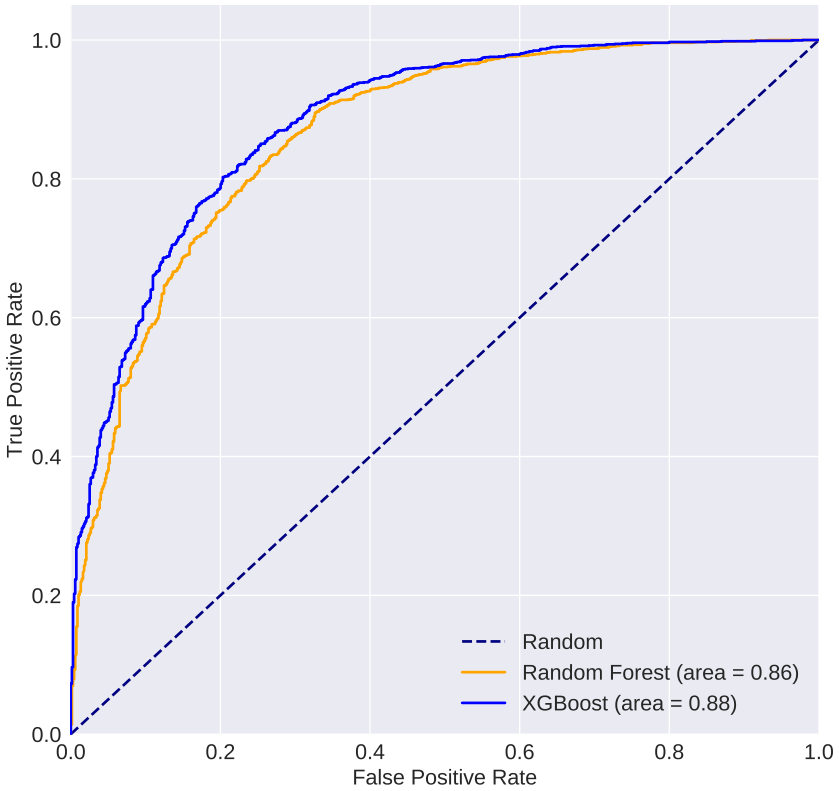


(a) Resultados del modelo usando RF. **(b)** Resultados del modelo usando XGB.

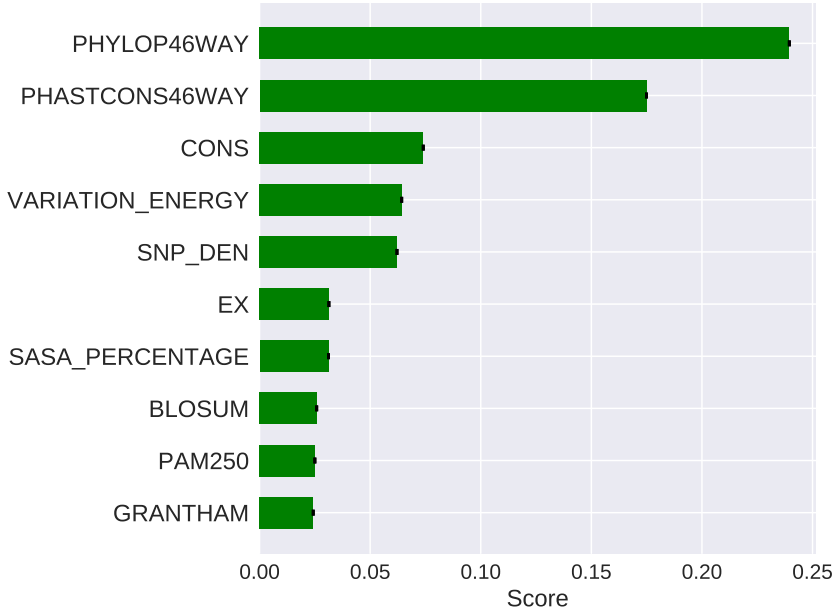
Fig. 6.2: Importancia de variables altamente correlacionadas del dataset Integral+VarQ Curado (basados en correlación de Spearman) usando permutación.

6.5. Conclusión del capítulo

La creación de un modelo combinando los datos de VarQ Curado con los del dataset Integral no superó lo obtenido por éste, aunque creemos que es posible mejorar este resultado calculando tanto las variables más importantes de VarQ (VARIATION_ENERGY y SASA) para las variantes del dataset Integral, como las variables genómicas para las variantes del dataset VarQ Curado.



(a) Curva AUC de los modelos Random Forest y XGBoost. La línea punteada corresponde a un predictor Random.



(b) Los 10 atributos más importantes del modelo Random Forest.

Fig. 6.3: Curva AUC y atributos más importantes del dataset Integral+VarQ Curado.

7. CONCLUSIONES GENERALES Y TRABAJO FUTURO

Al comienzo de este trabajo nos propusimos generar un modelo de predicción de patogenicidad en polimorfismos de un sólo nucleótido (SNPs), tratando de mejorar y aumentar lo conseguido a partir del análisis del modelo realizado con las variables que nos aportó VarQ. Con el objetivo de responder la primera pregunta formulada inicialmente, referida a si es posible enriquecer el dataset de VarQ compuesto esencialmente de variables de tipo estructural con variables de otras dimensiones (físico-químicas, genómicas, filogenéticas), buscamos trabajos de relevancia sobre el tema que nos permitieran incorporar una gran cantidad de variables nuevas, separadas en distintos grupos, según su interpretación biológica. Combinando estas variables con algoritmos de aprendizaje automático logramos una performance significativa en la identificación de variantes patogénicas, llegando a un AUC de 0.90 usando la totalidad de los SNPs reportados en Humsavar y a un AUC de 0.88 cuando evaluamos el modelo en el conjunto de variantes disponibles en el dataset de VarQ (ver figuras 7.1a y 7.1b).

El análisis estadístico de las variables de nuestro conjunto de datos nos permitió entender cómo el método estándar provisto en `scikit-learn` para calcular importancia de variables en métodos de ensamble se ve sesgado con la presencia de colinealidades en el conjunto de datos. En este sentido, el uso del método rfpimp [43], como método alternativo de análisis de importancia de variables facilitó la interpretación biológica de los resultados obtenidos. En particular, esto nos ayudó a responder nuestra segunda pregunta formulada en la introducción, referida a cómo afectan las distintas variables a nuestros modelos de predicción de patogenicidad, al identificar variables de importancia en cada una de las dimensiones estudiadas. En el caso del modelo VarQ Curado, encontramos como variables relevantes a la variación de la energía (VARIATION_ENERGY) y a la superficie accesible por parte del solvente (SASA). Luego, analizando las propiedades físico-químicas del cambio de aminoácido en la proteína, las variables más importantes fueron ciertas matrices de sustitución (EX, PAM250, BLOSUM, etc.), aunque con un grado muy alto de correlación entre ellas, la diferencia en la aromaticidad (AROMATICITY_DIFF) y la diferencia de hidrofobicidad (HYDROPHOBICITY) en la sustitución del aminoácido. Por último, en el caso genómico encontramos las variables con AUC univariado más elevado del trabajo, ambas referidas a la conservación filogenética del lugar (o el entorno) donde se produjo la variante: PHYLOP46WAY y PHASTCONS46WAY. Si bien el dataset de VarQ ya poseía una variable de conservación (CONS), ésta presentó un bajo nivel de relevancia en los modelos generados, el cual conjeturamos que se debe al origen de cálculo de esta variable: la misma se calcula a nivel de dominios funcionales Pfam, y con una familia mucho mayor de especies involucrada, sumado a su muy baja cobertura en los datasets estudiados.

Este trabajo también consistió en una comparación de distintos métodos de aprendizaje automático, buscando responder nuestro último interrogante: ¿Cuáles son los mejores algoritmos de aprendizaje automático para resolver este tipo de problemas y cuáles son sus hiperparámetros asociados?. En la primera sección del desarrollo comparamos tres métodos clásicos con el dataset VarQ Curado: Regresión Logística, Support Vector Classifier y Random Forest, obteniendo mejores resultados con éste último. Luego en la última sección del trabajo comparamos Random Forest con un método de *boosting*, XGBoost, en los datasets Integral y VarQ+Integral, mejorando la performance del modelo de un AUC de 0.88 a 0.90 y de 0.86 a 0.88 respectivamente, diferencias que mostraron ser estadísticamente significativas por el test de DeLong [21]. Si bien el tiempo de entrenamiento de los algoritmos fue relativamente bajo, esto se debe en

parte a haber acotado al subconjunto de hiperparámetros evaluados, por lo que hipotetizamos que todavía no se ha alcanzado el máximo nivel de desempeño posible de nuestro modelo.

Por último, cabe recalcar que uno de los principales productos del esfuerzo realizado en este trabajo es la integración de un nuevo dataset que contiene numerosas variables estructurales y físico-químicas de la proteína, sumadas a variables de tipo genómico que servirán como punto de partida para trabajos futuros del grupo. La combinación de estas variables demostró no ser superflua, mostrando una clara diferencia de performance entre el modelo Genómico (el de mejor desempeño independiente) y el Integral que contiene la sinergia de las distintas clases de variables exploradas.

7.1. Trabajo Futuro

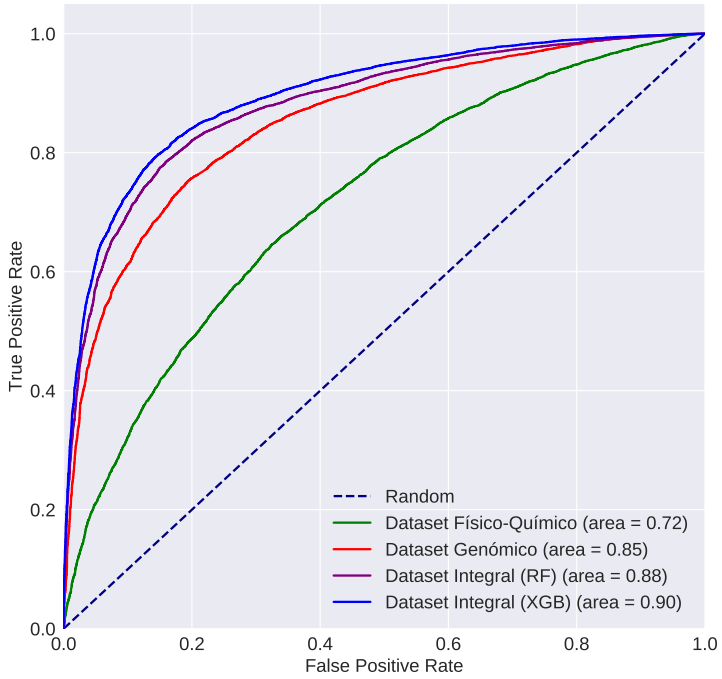
Una de las principales tareas que quedaron pendientes en este trabajo fue el cómputo de variables de VarQ para las variantes de Humsavar. Al mismo tiempo la intersección del dataset Integral con VarQ dió como resultado una cobertura baja de las variables de conservación genómicas, por lo que también es necesario calcular estas variables para todas las variantes.

Otro punto importante, como mencionamos en las conclusiones, es el espacio de búsqueda así como la técnica de selección de hiperparámetros para mejorar el desempeño de los algoritmos usados. Una las tareas posibles en este sentido es la implementación de un método más efectivo en la búsqueda de hiperparámetros óptimos, por ejemplo usando técnicas de optimización bayesiana. Por otro lado, la metodología de imputación de nulos en variables categóricas también admite el uso de enfoques más avanzados al utilizado en este trabajo. Otra posible dirección de exploración es la incorporación de otros algoritmos de aprendizaje automático, como por ejemplo el uso de redes neuronales o métodos bayesianos de inferencia.

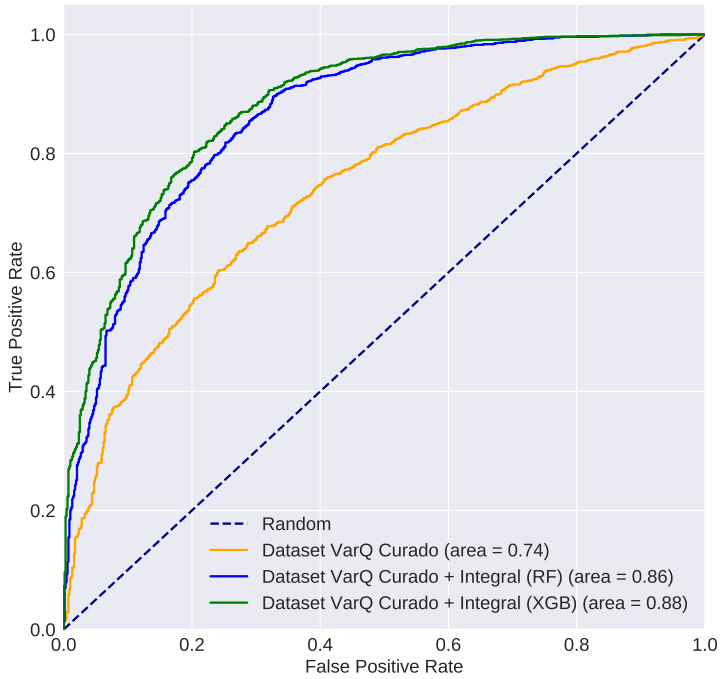
Además de las mejoras posibles al modelo en términos de desempeño (AUC), creemos que hay puntos metodológicos que pueden ser modificados y mejorados. El trabajo no presenta una descripción estadística referida a la cantidad de variantes por proteína, por lo tanto no sabemos si tenemos algún sesgo al entrenar y evaluar con variantes ocurridas en la misma proteína. Esto incurriría en un tipo de razonamiento circular de tipo 2 según el trabajo de Grimm et al. (2015) [52].

Otra de las modificaciones posibles es la extensión a SNPs de otros tipos, como por ejemplo variantes que generan un codón de terminación o *nonsense*. Esto otorgaría mayor relevancia a las variables de clase funcional incorporadas en el dataset genómico, que por el momento no son aprovechadas dado que la totalidad de nuestras variantes son de clase *missense*.

Los datasets utilizados poseen información adicional que no fue utilizada en nuestro trabajo, por ejemplo la enfermedad asociada a la variante. Utilizando esta información es posible restringirse a algún tipo de enfermedad particular, o reconocer clases de enfermedades, pasando de un problema de clasificación binario a uno de múltiples clases.



(a) Comparación de curvas ROC entre los datasets Físico-Químico, Genómico e Integral.



(b) Comparación de curvas ROC entre los datasets VarQ Curado y VarQ Curado + Integral.

Fig. 7.1: Comparación de curvas AUC usando datasets con variantes de Humsavar y VarQ Curado.

Bibliografía

[1] Franklin, R. y R. Gosling: *Molecular Configuration in Sodium Thymonucleate*. Nature, 171:740, apr 1953. https://doi.org/10.1038/171740a0.

[2] Watson, J. D. y F. H. C. Crick: *Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid*. Nature, 171:737, apr 1953. http://dx.doi.org/10.1038/171737a0.

[3] International Human Genome Sequencing Consortium et al.: *Initial sequencing and analysis of the human genome*. Nature, 409:860, feb 2001. http://dx.doi.org/10.1038/35057062.

[4] The ENCODE Project Consortium et al.: *An integrated encyclopedia of DNA elements in the human genome*. Nature, 489:57, sep 2012. http://dx.doi.org/10.1038/nature11247.

[5] Hamburg, Margaret A. y Francis S. Collins: *The Path to Personalized Medicine*. New England Journal of Medicine, 363(4):301–304, 2010. https://doi.org/10.1056/NEJMp1006304.

[6] Crick, F. H. C.: *On Protein Synthesis*. Symposia of the Society for Experimental Biology. Cambridge University Press., páginas 138–163, 1958.

[7] Crick, F. H. C.: *Central Dogma of Molecular Biology*. Nature, 227:561, aug 1970. http://dx.doi.org/10.1038/227561a0.

[8] *The Genetic Code (OpenStax CNX)*. https://cnx.org/contents/GFy_h8cu@9.87:QEibhJMi@8/The-Genetic-Code. Visitado: 15-10-2018.

[9] *Human Genetic Variation: An Introduction (EMBL-EBI)*. https://www.ebi.ac.uk/training/online/course/human-genetic-variation-i-introduction. Visitado: 17-10-2018.

[10] KA., Wetterstrand: *DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP)*. www.genome.gov/sequencingcostsdata. Visitado: 02-02-2019.

[11] *Human polymorphisms and disease mutations*. https://www.uniprot.org/docs/humsavar. Visitado: 20-12-2017.

[12] *ClinVar*. https://www.ncbi.nlm.nih.gov/clinvar/. Visitado: 20-01-2019.

[13] Barber, D.: *Bayesian Reasoning and Machine Learning*. Machine Learning, página 646, 2011, ISSN 9780521518147.

[14] Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot y E. Duchesnay: *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12:2825–2830, 2011.

[15] *Machine Learning Challenge Winning Solutions*. https://github.com/dmlc/xgboost/tree/master/demo#machine-learning-challenge-winning-solutions. Visitado: 19-01-2019.

[16] Hastie, T., R. Tibshirani y J. Friedman: *The Elements of Statistical Learning*, volumen 27. Springer, 2001, ISBN 978-0-387-84857-0.

[17] *How to explain Gradient Boosting*. `https://explained.ai/gradient-boosting/index.html`. Visitado: 20-01-2019.

[18] Fawcett, T.: *An introduction to ROC analysis*. Pattern Recognition Letters, 27(8):861–874, 2006, ISSN 01678655.

[19] Mason, S. J. y N. E. Graham: *Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation*. Quarterly Journal of the Royal Meteorological Society, 128(584):2145–2166, 2002. `https://rmets.onlinelibrary.wiley.com/doi/abs/10.1256/003590002320603584`.

[20] Mann, H. B. y D. R. Whitney: *On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other*. Ann. Math. Statist., 18(1):50–60, Marzo 1947. `https://doi.org/10.1214/aoms/1177730491`.

[21] DeLong, Elizabeth R., David M. DeLong y Daniel L. Clarke-Pearson: *Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach*. Biometrics, 44(3):837–845, 1988, ISSN 0006341X, 15410420. `http://www.jstor.org/stable/2531595`.

[22] Radusky, L.: *Herramientas bioinformáticas para el análisis estructural de proteínas a escala genómica*. Tesis de Doctorado, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, 2017.

[23] Carter, H., C. Douville, P. Stenson, D. Cooper y R. Karchin: *Identifying Mendelian disease genes with the Variant Effect Scoring Tool*. BMC Genomics, 14(Suppl 3):S3, 2013, ISSN 1471-2164. `http://bmcgenomics.biomedcentral.com/articles/10.1186/1471-2164-14-S3-S3`.

[24] Shihab, H. A., M. F. Rogers, J. Gough, M. Mort, D. N. Cooper, I.N.M. Day, T. R. Gaunt y C. Campbell: *An integrative approach to predicting the functional effects of non-coding and coding sequence variation*. Bioinformatics, 31(10):1536–1543, 2015, ISSN 14602059.

[25] *VarQ: Structural analysis of protein variants*. `http://varq.qb.fcen.uba.ar/`. Visitado: 20-01-2019.

[26] Schymkowitz, J., J. Borg, F. Stricher, R. Nys, F. Rousseau y L. Serrano: *The FoldX web server: an online force field*. Nucleic acids research, 33(Web Server issue):W382–W388, jul 2005, ISSN 1362-4962. `https://www.ncbi.nlm.nih.gov/pubmed/15980494`.

[27] Diaz, C., H. Corentin, V. Thierry, A. Chantal, B. Tanguy, S. David, H. Jean-Marc, F. Pascual, B. Françoise y F. Edgardo: *Virtual screening on an α-helix to β-strand switchable region of the FGFR2 extracellular domain revealed positive and negative modulators*. Proteins, 82(11):2982–2997, 2014, ISSN 0887-3585. `https://doi.org/10.1002/prot.24657`.

[28] Fernandez-Escamilla, A., F. Rousseau, J. Schymkowitz y L. Serrano: *Prediction of sequence-dependent and mutational effects on the aggregation of peptides and proteins*. Nature Biotechnology, 22(10):1302–1306, 2004, ISSN 10870156.

[29] Finn, R. D., A. Bateman, J. Clements, P. Coggill, R. Y. Eberhardt, S. R. Eddy, A. Heger, K. Hetherington, L. Holm, J. Mistry, E. L. L. Sonnhammer, J. Tate y M. Punta: *Pfam: the protein families database*. Nucleic acids research, 42(Database issue):D222–D230, jan 2014, ISSN 1362-4962. `https://www.ncbi.nlm.nih.gov/pubmed/24288371`.

[30] Porter, C. T., G. J. Bartlett y J. M. Thornton: *The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data*. Nucleic acids research, 32(Database issue):D129–D133, jan 2004, ISSN 1362-4962. `https://www.ncbi.nlm.nih.gov/pubmed/14681376`.

[31] Stein, A., R. B. Russell y P. Aloy: *3did: interacting protein domains of known three-dimensional structure*. Nucleic acids research, 33(Database issue):D413–D417, jan 2005, ISSN 1362-4962. `https://www.ncbi.nlm.nih.gov/pubmed/15608228https://www.ncbi.nlm.nih.gov/pmc/PMC539991/`.

[32] Berman, H., K. Henrick y H. Nakamura: *Announcing the worldwide Protein Data Bank*. Nature Structural Biology, 10:980, dec 2003. `https://doi.org/10.1038/nsb1203-980http://10.0.4.14/nsb1203-980`.

[33] Brodersen, K. H., C. S. Ong, K. E. Stephan y J. M. Buhmann: *The balanced accuracy and its posterior distribution*. Proceedings - International Conference on Pattern Recognition, páginas 3121–3124, 2010, ISSN 10514651.

[34] Chapman, B. y J. Chang: *Biopython: Python Tools for Computational Biology*. SIGBIO Newsl., 20(2):15–19, Agosto 2000, ISSN 0163-5697. `http://doi.acm.org/10.1145/360262.360268`.

[35] Wong, W. C., D. Kim, H. Carter, M. Diekhans, M. C. Ryan y R. Karchin: *CHASM and SNVBox: Toolkit for detecting biologically important single nucleotide mutations in cancer*. Bioinformatics, 27(15):2147–2148, 2011, ISSN 13674803.

[36] *Biopython License Agreement*. `https://github.com/biopython/biopython/blob/master/LICENSE.rst`. Visitado: 20-12-2017.

[37] *ProtParam tool*. `https://web.expasy.org/protparam/`. Visitado: 20-12-2017.

[38] Vihinen, M., E. Torkkila y P. Riikonen: *Accuracy of protein flexibility predictions*. Proteins: Structure, Function, and Bioinformatics, 19(2):141–149, 1994. `https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.340190207`.

[39] The UniProt Consortium: *UniProt: the universal protein knowledgebase*. Nucleic Acids Research, 45(D1):D158–D169, 2017. `http://dx.doi.org/10.1093/nar/gkw1099`.

[40] Goldstein, Tom, Christoph Studer y Richard Baraniuk: *A Field Guide to Forward-Backward Splitting with a FASTA Implementation*. arXiv eprint, abs/1411.3406, 2014. `http://arxiv.org/abs/1411.3406`.

[41] Gasteiger, Elisabeth, Christine Hoogland, Alexandre Gattiker, S'everine Duvaud, Marc R. Wilkins, Ron D. Appel y Amos Bairoch: *Protein Identification and Analysis Tools on the ExPASy Server*. The Proteomics Protocols Handbook, páginas 571–607, 2005, ISSN 10643745. `http://link.springer.com/10.1385/1-59259-890-0:571`.

[42] Buske, O. J., A. Manickaraj, S. Mital, P. N. Ray y M. Brudno: *Identification of deleterious synonymous variants in human genomes*. Bioinformatics, 29(15):1843–1850, 2013. http://dx.doi.org/10.1093/bioinformatics/btt308.

[43] *Feature importances for scikit random forests*. https://github.com/parrt/random-forest-importances. Visitado: 20-12-2017.

[44] Altmann, A., L. Tolo, O. Sander y T. Lengauer: *Permutation importance: a corrected feature importance measure*. Bioinformatics, 26(10):1340–1347, 2010.

[45] *dbSNP Short Genetic Variations*. https://www.ncbi.nlm.nih.gov/projects/SNP/. Visitado: 21-10-2018.

[46] Siepel, A., G. Bejerano, J.S. Pedersen, A.S. Hinrichs, M. Hou, K. Rosenbloom, H. Clawson, J. Spieth, L.D.W. Hillier, S. Richards *y cols.*: *Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes*. Genome research, 15(8):1034, 2005.

[47] Pollard, K. S., M. J. Hubisz, K. R. Rosenbloom y A. Siepel: *Detection of nonneutral substitution rates on mammalian phylogenies*. Genome Research, páginas 110–121, 2010.

[48] Karolchik, D.: *The UCSC Table Browser data retrieval tool*. Nucleic Acids Research, 32(90001):493D–496, 2004, ISSN 1362-4962. https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkh103.

[49] Kitts, A., L. Phan, M. Ward y J. Bradley: *The Database of Short Genetic Variation (dbSNP)*. En *NCBI Handbook*. Bethesda (MD): National Center for Biotechnology Information (US), 2013/06/30 edición, 2014. https://www.ncbi.nlm.nih.gov/books/NBK174586/.

[50] Kitts, A., L. Phan, M. Ward y J. Bradley: *Table 4. [Molecular codes for refSNPs in gene features]*. https://www.ncbi.nlm.nih.gov/books/NBK174586/table/dbSNP.T.molecular_codes_for_refsnps_in_g/. Visitado: 03-02-2019.

[51] Ostell, J. y J. McEntyre: *The NCBI Handbook*. NCBI Bookshelf, páginas 1–8, 2007. www.ncbi.nlm.nih.gov/books/NBK21101/?report=printable.

[52] Grimm, D. G., C. Azencott, F. Aicheler, U. Gieraths, D. G. MacArthur, K. E. Samocha, D. N. Cooper, P. D. Stenson, M. J. Daly, J. W. Smoller, L. E. Duncan y K. M. Borgwardt: *The Evaluation of Tools Used to Predict the Impact of Missense Variants Is Hindered by Two Types of Circularity*. Human Mutation, 36(5):513–523, 2015. https://onlinelibrary.wiley.com/doi/abs/10.1002/humu.22768.

[53] *SNVBox Features*. http://wiki.chasmsoftware.org/images/4/4d/SNVBox_Final.pdf. Visitado: 14-02-2019.

7. APÉNDICE

7.2. Estructura del proyecto

A continuación detallamos los distintos módulos del proyecto para facilitar la replicación de los resultados obtenidos. Este se encuentra alojado en GitHub en el sitio `https://github.com/marlanbar/master-thesis`.

El proyecto tiene la siguiente estructura:

```
master-thesis
├── data
│   ├── external
│   ├── interim
│   └── processed
├── notebooks
├── results
│   ├── varq
│   ├── physico_chemical
│   ├── genomic
│   ├── integral
│   └── varq_integral
└── src
    └── snvbox_queries
```

En `src` se encuentra el código necesario para obtener las variables de la tabla SNVbox y generar las variables a partir de la data externa (por ejemplo, Humsavar) que se encuentra en la carpeta `data/external`. Las variables procesadas se encuentran en `data/interim`. Luego, para cada modelo existe una notebook para generar los datasets, ubicados en `data/processed`, y otro para evaluar los algoritmos, que almacenan los scores de las variantes en cada una de sus respectivas carpetas en `results`. El trabajo fue realizado con una notebook Dell XPS 9350 (Intel Core i5 Skylake, 8GB de memoria RAM, disco SSD 128GB).

7.3. Diccionario de hiperparámetros usados

Para los modelos usados, se usaron los siguientes diccionarios de parámetros:

- Random Forest
  - Profundidad del árbol (`max_depth`): [3, 5, 7]
  - Estimadores (`n_estimators`): [10, 50, 100]
  - Cantidad de máxima de variables por árbol (`max_features`): [4, $\sqrt{n}$, 0.2*n] con n la cantidad total de variables
- Regresión logística:
  - Parámetro de regularización inverso (C): [.001, .01, .1, 1, 10, 100, 1000]
  - Peso de las clases: [balanceado, igual a 1]

- SVC:
  - Parámetro de penalidad (C): [0.001, 0.10, 0.1, 10, 25, 50, 100, 1000]
  - Gamma: [0.001, 0.10, 0.1, 10, 25, 50, 100, 1000]
- XGBoost:
  - Peso mínimo de las hojas: [1, 5, 10]
  - Gamma: [0.5, 1, 1.5, 2, 5]
  - Subsample: [0.6, 0.8, 1.0]
  - colsample_bytree: [0.6, 0.8, 1.0]
  - Profundidad máxima: [3, 4, 5]

7.4. Lista de Variables de SNVBox

La siguiente sección es una lista detallada de variables de SNVBox que fueron usadas en el trabajo. La lista completa junto con el esquema de la base de datos se encuentra en la wiki del proyecto SNVBox [53].

Variables sobre cambios en la sustitución del Aminoácido (Estructura Primaria)

- Score BLOSUM (AABLOSUM): Score de la matriz BLOSUM 62.
- Carga (AACharge): El cambio en la carga resultante de cambiar el aminoácido de referencia con la mutación.
- Volumen (AAVolume): El cambio en el residuo resultante del cambio de aminoácido (expresado en Angstroms cúbicos).
- Hidrofobia (AAHydrophobicity): El cambio en hidrofobicidad resultante de la mutación.
- Score Grantham (AAGrantham): La distancia Grantham de la referencia a la mutación.
- Polaridad (Polarity): Cambio de polaridad entre la referencia y la mutación.
- Score Ex (AAEx): Score de la matriz EX.
- Score PAM250 (AAPAM250): Score de la matriz PAM250.
- Score MJ (AAMJ): Score de la matriz Miyagawa-Jerningan.
- Score VB (AAVB): Score de la matriz VB (Venkatarajan & Braun).
- Transición (Transition): Frecuencia de la transición entre dos aminoácidos vecinos basado en todas las proteínas de SwissProt/TrEMBL.

Variables a nivel de Proteína (sin considerar sustitución)

- BINDING: Sitio de unión.
- ACTIVE_SITE: Actividad enzimática.
- SITE: Sitio de aminoácido "interesante" en la secuencia proteica.
- LIPID: Unión con un lípido.
- METAL: Unión con un metal.
- CARBOHYD: Unión con un carbohidrato.
- DNA_BIND: Unión con ADN.
- NP_BIND: Unión con un nucleótido fosfato.
- CA_BIND: Unión con calcio.
- DISULFID: Sitio de unión con un disulfuro.
- SE_CYS: Selenocisteína.
- MOD_RES: Residuo modificado.
- PROPEP: Propéptido.
- SIGNAL: Sitio de señal de localización.
- TRANSMEM: Proteína transmembranal.
- COMPBIAS: Sesgo de composición.
- REP: Región de repetición.
- MOTIF: Sitio de un motif funcional conocido.
- ZN_FING: Dedo de zinc.
- REGIONS: Región de interés en la secuencia proteica.
- PPI: Interacción proteína-proteína.
- RNABD: Unión RNA.
- TF: Factor de transcripción.
- LOC: Sitio que determina correcta localización celular de una proteína.
- MMBRBD: Sitio que se une a la membrana celular.