UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE CIENCIAS BIOLÓGICAS

DESARROLLO DE UN MÉTODO BASADO EN APRENDIZAJE AUTOMÁTICO PARA LA PREDICCIÓN DEL CAMBIO DE AFINIDAD POR MUTACIONES PUNTUALES EN COMPLEJOS ANTÍGENO-ANTICUERPO

VÍCTOR IGNACIO FICA LEÓN

Tesis para optar al grado de Magíster en Bioquímica y Bioinformática

Profesor Guía: Dr. Alexis Marcelo Salas Burgos
Dpto. de Farmacología, Facultad de Ciencias Biológicas
Universidad de Concepción

Concepción, Chile 2021

AGRADECIMIENTOS

Quiero agradecer a mi familia, por el entendimiento, apoyo y paciencia que han tenido conmigo todos estos años.

A mi profesor, mentor y amigo Alexis Salas Burgos, por integrarme y aceptarme en el ámbito académico y personal. Por sus enseñanzas y constante apoyo, y su inagotable buena disposición.

A todos mis compañeros de laboratorio, los que han pasado y han llegado durante estos años, por la camaradería y la buena disposición.

Y finalmente a mis compañeros de carrera, bioingeniería generación 2009, por la inagotable amistad que perdura más allá de la Universidad.

Gracias a todos y que la fuerza los acompañe.

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ABREVIATURAS

Aminoácidos

Una letra	Código 3 letras	Aminoácidos
A	Ala	Alanina
C	Cys	Cisteína
D	Asp	Ácido aspártico
E	Glu	Ácido glutámico
F	Phe	Fenilalanina
G	Gly	Glicina
H	His	Histidina
I	Ile	Isoleucina
K	Lys	Lisina
L	Leu	Leucina
M	Met	Metionina
N	Asn	Asparagina
P	Pro	Prolina
Q	Gln	Glutamina
R	Arg	Arginina
S	Ser	Serina
T	Thr	Treonina
V	Val	Valina
W	Trp	Triptófano
Y	Tyr	Tirosina

Misceláneos

AA : aminoacid
Ab : antibody
Ag : antigen
CDR : complementary determining region
Fab : antigen-binding fragment

Fc	: fragment crystallizable
FDA	: food and drug administration
FR/FW	: framework
Fv	: variable domain antibody fragment
mAb	: monoclonal antibody
RMSE	: root mean squared error
SASA	: solvent accessible surface area
SHM	: somatic hypermutations
SVR	: support vector regressor
GBT	: gradient boosted trees
PCC	: pearsons`s correlation coefficient
MAE	: mean absolute error
PPI	: protein-protein interactions
V_H	: heavy chain variable domain
V_L	: light chain variable domain

RESUMEN

Los anticuerpos son una de las moléculas más importantes en la investigación de biofármacos. Esto ha impulsado el desarrollo de técnicas experimentales para diseñar anticuerpos (Ab) de alta especificidad y afinidad por su antígeno. Sin embargo, estos métodos poseen sus limitantes, y cada vez es más claro que los métodos de diseño computacional de anticuerpos facilitan este proceso.

Actualmente aún existe la necesidad de desarrollar métodos computacionales fiables y rápidos que permitan estimar los efectos de mutaciones en la afinidad de unión (es decir, el cambio de energía libre de unión entre un complejo nativo y mutante, ΔΔG). Métodos computacionales de tales características pueden ser de gran ayuda en procesos como la maduración de la afinidad *in silico* de anticuerpos, donde la combinatoria de sustituciones en los CDR puede alcanzar fácilmente el orden de 20^{60} .

La creciente cantidad de secuencias e información estructural ha centrado el interés en la aplicación de aprendizaje automático para el desarrollo de herramientas predictivas en múltiples áreas de la bioinformática. En tareas como la predicción de afinidad de anticuerpos han demostrado superar a métodos clásicos, como las funciones empíricas o las basadas en conocimiento. Sin

embargo, existen pocos métodos computacionales que aborden directamente el problema de la maduración de la afinidad *in silico* de anticuerpos, de tal manera que permita hacer menos costoso computacionalmente este proceso.

Este trabajo abordó el desarrollo de un nuevo método basado en aprendizaje automático específico para complejos Antígeno-Anticuerpo capaz de predecir el cambio de afinidad por mutaciones puntuales llamado ***ABPRED***. Para ello se obtuvo una base de datos de constantes cinéticas para mutantes simples en complejos Antígeno-Anticuerpo. Se modelaron las mutantes y retromutantes, para posteriormente calcular características estructurales y energéticas con las cuales entrenar y evaluar un modelo. Finalmente se comparó ABPRED frente a otros predictores representativos actualmente disponibles como FoldX, BeatMusic, DFIRE2 y mCSM-AB.

ABPRED alcanzó una correlación de Pearson de 0.6 (RMSE = 1.457) y 0.55 (RMSE = 1.49), en el conjunto de entrenamiento y prueba respectivamente, demostrando un mejor rendimiento a métodos clásicos que han sido previamente usados para diseñar anticuerpos.

ABSTRACT

Antibodies are one of the most important molecules in biopharmaceutical research. This has prompted the development of experimental techniques to design antibodies (Ab) with high specificity and affinity for their antigen. However, these methods have their limitations, and it is becoming increasingly clear that methods for computational antibody design facilitate this process.

There is still a need to develop reliable and fast computational methods that allow estimating the effects of mutations on binding affinity (ie, changes in binding free energy between a wildtype and mutant complex, ΔΔG). Computational methods of such characteristics can help in processes such as *in silico* affinity maturation of antibodies, where CDRs substitutions can easily reach a combinatorial of 20^{60}.

The increasing amount of structural and sequence information has put the interest in the application of machine learning for the development of predictive tools in multiple bioinformatic areas. In tasks such as antibody affinity prediction, they have been shown to surpass classical methods, such as empirical or knowledge-based functions. However, there are few computational methods that directly address the problem of *in silico* affinity maturation of antibodies, making this process less computationally expensive.

In this work we develop a new machine learning based method specific for Antigen-Antibody complexes capable of predicting affinity changes upon point mutations called ABPRED. For this, a database of kinetic constants for single point mutations on Antigen-Antibody complexes was obtained. Mutants and retromutants were modeled, then structural and energetic features were defined to train and evaluate a model. Finally, ABPRED was compared against other currently available representative predictors such as FoldX, BeatMusic, DFIRE2 and mCSM-AB.

ABPRED reached a Pearson`s correlation of 0.6 (RMSE = 1.457) and 0.55 (RMSE = 1.49), in the training and test set respectively, showing better performance than previously used classical methods for antibody design.

1 INTRODUCCIÓN

Los métodos computacionales se han convertido en una herramienta esencial para el desarrollo de fármacos, facilitando la búsqueda de compuestos, evaluación de la “drogabilidad”, y la optimización de propiedades fisicoquímicas y farmacocinéticas (Talevi, 2018). Estos métodos permiten generar hipótesis evaluables, ayudando a interpretar y guiar los experimentos. Este enfoque de investigación ha recibido particular interés en el diseño y desarrollo de un tipo especial de biofármaco: los anticuerpos de uso terapéutico.

1.1 Anticuerpos Terapéuticos

Los anticuerpos son una de las moléculas más importantes en la investigación de biofármacos, con un creciente interés como uso terapéutico en diferentes enfermedades humanas. En la actualidad existen 78 anticuerpos monoclonales (mAb) aprobados por la Food and Drug Administration (FDA) (Antibody Society 2019), y otro gran número está en etapa clínica avanzada o esperando aprobación para entrar al mercado (Kaplon and Reichert 2019). Los anticuerpos monoclonales poseen ciertas ventajas frente a otras proteínas terapéuticas, tales como: larga vida media en suero, mayor avidez y selectividad, y la capacidad de

activar respuestas inmunes deseadas (Chames et al. 2009). El paratopo, la región del anticuerpo que interacciona con el antígeno, puede reconocer prácticamente cualquier biomolécula objetivo, con amplio rango de afinidades y especificidades. Esta versatilidad de unión significa que pueden ser diseñados contra un epítopo arbitrario, el motivo estructural del antígeno, por lo cual han sido el centro de atención de la industria farmacéutica. Los anticuerpos de uso terapéutico pueden ser desarrollados, madurados y aislados con diferentes métodos. Los principales incluyen: la tecnología de hibridoma (Köhler and Milstein 1975), humanización de anticuerpos (Almagro and Fransson 2008), ratones transgénicos “humanizados” (Jakobovits 1995), y presentación sobre fagos (Bradbury et al. 2011). En la actualidad, la mayoría de los procesos de generación de anticuerpos terapéuticos utilizan métodos computacionales para guiar el diseño, centrándose en aspectos como el aumento de la afinidad (Clark et al. 2006), control de especificidad (Farady et al. 2009), diseño de termoestabilidad (McConnell et al. 2013), propensidad de agregación (Lauer et al. 2012) y en los mismos procesos de humanización (Choi et al. 2015).

1.2 Características y Estructura del Anticuerpo

Un anticuerpo es una molécula de proteína presente en el plasma y líquidos intersticiales, generada por las células B diferenciadas en respuesta a la entrada de un antígeno. El anticuerpo así generado puede ejercer su función efectora

tales como: la citotoxicidad celular dependiente de anticuerpos (en inglés ADCC), citotoxicidad dependiente del complemento (en inglés CDC), la opsonización y fagocitosis, y la neutralización (Basic Immunology - 6th Edition).

Los anticuerpos nativos son glicoproteínas hetero-tetraméricas de 150 kDa con cuatro cadenas polipeptídicas: dos cadenas pesadas idénticas (H) y dos cadenas ligeras idénticas (L). Como se observa en la Figura 1, las cadenas ligeras y pesadas están unidas por puentes disulfuro formando los brazos de una estructura en forma de Y; cada brazo es conocido como un Fab (del inglés Fragment, *antigen binding*). El Fab se compone de dos dominios variables de las cadenas pesadas y ligeras (V_H y V_L) y dos dominios constantes (C_H y C_L). En el emparejamiento de las cadenas ligeras y pesadas, los dos dominios variables dimerizan para formar el fragmento Fv (variable domain antibody fragment) que contiene el sitio de unión al antígeno. Dentro de cada dominio variable se encuentran seis lazos hipervariables (Wu and Kabat 1970) (CDRs, regiones determinantes de la complementariedad), tres en la cadena ligera (L1, L2 y L3) y tres en la cadena pesada (H1, H2 y H3), flanqueadas por regiones altamente conservadas de estructura en hojas beta, llamadas regiones marco o de entramado (FR/FW, o *frameworks*). Los dominios variables ligeros y pesados se pliegan de una manera en la cual los lazos hipervariables se juntan para crear el sitio de unión al antígeno o paratopo. Dos dominios adicionales de la cadena pesada, C_H2, y C_H3, forman la región Fc (*Fragment crystallizable*) que es responsable de mediar la actividad efectora de la molécula de anticuerpo. La

comparación de estructuras cristalográficas de anticuerpos ha revelado que la conformación de los CDRs, a excepción del H3, son relativamente rígidos y pueden ser clasificados en las llamadas estructuras canónicas (North, Lehmann, and Dunbrack Jr 2011). La estructura modular de los anticuerpos ha sido aprovechada en su diseño, así métodos como la humanización de anticuerpos se basan en injertar CDRs no humanos en regiones FR humanas (Almagro and Fransson 2008).

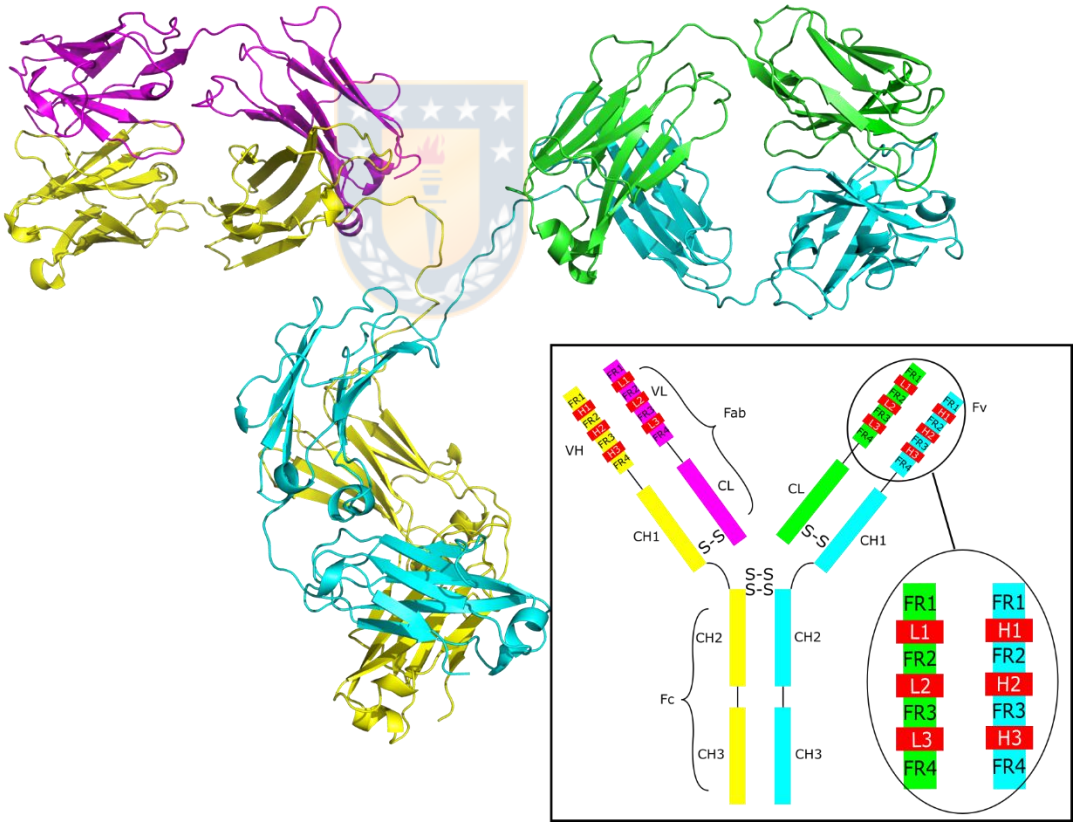

Figura 1. Representación esquemática de un anticuerpo.
Un anticuerpo consta de 2 cadenas livianas y 2 pesadas unidas en forma de Y. Cada "brazo" del anticuerpo forma el fragmento de unión al antígeno (Fab). Los dominios variables de la cadena liviana y pesada forman el fragmento variable (Fv), en el cual se encuentran 6 lazos hipervariables llamados CDR. Adaptado desde (Sela-Culang, Kunik, and Ofran 2013)

1.2 Anticuerpos y el proceso de maduración

Las proteínas evolucionan con el tiempo a través de mutaciones aleatorias en sus secuencias genómicas. De manera similar, pero de forma más rápida, los anticuerpos evolucionan en respuesta a antígenos a través de mutaciones en sus líneas germinales. Esta evolución es guiada por hipermutaciones somáticas (en inglés SHM) y un proceso de selección, lo que resulta en la acumulación gradual de mutaciones por todo el anticuerpo, pero con mayor probabilidad en los CDRs, especialmente en la cadena pesada (Burkovitz, Sela-Culang, and Ofran 2014). Estas mutaciones hipersomáticas llevan a una mayor afinidad y mejor complementariedad en la interfaz antígeno-anticuerpo (Li et al. 2003). Una mayor comprensión de estos procesos es esencial para la búsqueda de mejores estrategias de inmunización, guiando los procesos de diseño (Doria-Rose and Joyce 2015). Los mecanismos de diversificación de anticuerpos han evolucionado para lograr un equilibrio entre la plasticidad necesaria para unirse con éxito a nuevos antígenos desconocidos y la robustez necesaria para hacerlo de una manera biológicamente viable. Esto da como resultado una serie de patrones y variaciones que pueden estudiarse computacionalmente para comprender los procesos celulares subyacentes y predecir la respuesta a manipulaciones específicas (Schramm and Douek 2018).

1.2.1 Maduración de la afinidad *in vitro*

Los anticuerpos también pueden evolucionar artificialmente *in vitro* utilizando tecnologías de presentación, como por ejemplo, la presentación sobre fagos (Bradbury et al. 2011). Estas tecnologías permiten la implementación de repertorios de genes específicos, logrando el desarrollo de anticuerpos sintéticos con propiedades deseadas (Finlay and Almagro 2012). La ventaja de estos métodos radica en que no necesitan inmunización animal para aislar el anticuerpo. Sin embargo, muchas veces se dificulta seleccionar epítopes específicos usando solo esta tecnología, por lo cual una ruta alternativa es combinarlos junto a modelamiento estructural computacional (Barderas et al. 2008).

1.4 Diseño computacional de proteínas

Los algoritmos enfocados en el diseño de proteínas han mejorado consistentemente, siendo capaces de revelar aspectos esenciales en la arquitectura de las proteínas (Koga et al. 2012). El diseño computacional de proteínas se basa en la estimación de los cambios en la afinidad de unión dado la sustitución de aminoácidos, lo cual puede ser conseguido utilizando funciones de puntuación, como los campos de fuerza o potenciales “basados en conocimiento” derivados desde base de datos estructurales (Moal et al. 2013). El

diseño de proteínas además requiere algoritmos de búsqueda conformacional eficientes, a menudo con bibliotecas de rotámeros de cadenas laterales, para muestrear las conformaciones de estructuras modificadas (Dunbrack 2002).

El diseño de proteínas puede ser clasificado en 2 categorías: rediseño y diseño *de novo.* Durante el rediseño, las funciones de puntuación y los algoritmos de búsqueda conformacional utilizan como punto de partida la información de una proteína nativa o una interfaz. La mayoría de los esfuerzos iniciales en diseño racional de proteínas involucra el rediseño de la superficie o interior de proteínas monoméricas, de manera que los monómeros existentes aumenten su termoestabilidad (Dahiyat and Mayo 1997). Más tarde, el diseño computacional extendió su habilidad para crear nuevos plegamientos no naturales (Kuhlman et al. 2003) y su capacidad para alterar interfaces proteína-proteína, de esta forma se logró madurar la afinidad de complejos proteicos (Selzer, Albeck, and Schreiber 2000).

1.5 Predicción del cambio en la afinidad de unión

Las interacciones proteína-proteína (protein-protein interactions, PPIs) son el centro de la mayoría de las funciones y actividades biológicas, por lo que el estudio de la información tridimensional de complejos proteicos es fundamental. Sin embargo, solo un ~13.3% del interactoma humano conocido posee información estructural (Interactome3D, version 2019_01) (Mosca, Céol, and Aloy

2013). Esto ha impulsado el desarrollo de métodos de modelamiento computacional complementarios que permitan suplir esta falta de información estructural en PPIs.

Además de adquirir conocimiento estructural, comprender la termodinámica de las PPIs es clave para dilucidar mecanismos de acción, comprender el efecto de mutaciones relacionadas a enfermedades y/o diseñar nuevas interacciones. Uno de los elementos termodinámicos más importantes que permite describir la fuerza de interacción entre proteínas es la afinidad de unión o energía libre de unión (ΔG). Los cambios en la energía libre de unión (ΔΔG) causados por mutaciones pueden mostrar el impacto de estas en las interacciones proteína-proteína. El ΔΔG está definido como la diferencia en la afinidad de unión entre el complejo mutante (*mutant*) y nativo (*wildtype*):

$$\Delta\Delta G = \Delta Gmt - \Delta Gwt \qquad \text{Ecuación 1.}$$

1.5.1 Predictores representativos de ΔΔG

En las últimas 2 décadas se han publicado varios predictores de ΔΔG basados en estructura (Figura 2). En general estos predictores se pueden clasificar en funciones lineales clásicas y métodos basados en aprendizaje automático. Las funciones lineales clásicas están basadas en energías físicas (o campos de fuerza) y/o potenciales estadísticos, cuyos pesos de cada término son

optimizados para reproducir los datos experimentales. En contraste, las funciones basadas en aprendizaje automático pueden usar una variedad de información estructural, evolutiva, energética, etc. para estimar el ΔΔG.

Los primeros predictores de ΔΔG, en general, usan campos de fuerzas como componentes descriptivos de las PPIs. FoldX (Guerois, Nielsen, and Serrano 2002) es un representativo de este tipo de predictores, el cual sigue siendo ampliamente utilizado para predecir el efecto de mutaciones. Está basado en energía físicas del tipo Van der Waals y electrostáticas, más la adición de enlaces hidrógeno y solvatación. Para modelar las mutaciones, FoldX utiliza la aproximación de rotámeros, permitiendo solo los cambios conformacionales de las cadenas laterales y dejando fija la cadena principal. El mismo año que FoldX fue publicado, un predictor de ΔΔG basado en Rosetta (Kortemme and Baker 2002), el que utilizaba una combinación lineal de energías físicas, principalmente interacciones Lennard Jones, interacciones de solvatación y enlaces hidrógeno. También utilizó la aproximación con rotámeros y estaba limitado a mutaciones de alanina. El método CC/PBSA (del inglés *Concoord/Poisson-Bolztmann surface area*) publicado en 2009 (Benedix et al. 2009), fue uno de los primeros predictores de ΔΔG en utilizar ensambles estructurales para describir explícitamente los cambios conformacionales tanto de la cadena principal y lateral. Este método estima el ΔΔG utilizando energías físicas ponderadas sobre todas las estructuras minimizadas de un ensamble, sin embargo, el cálculo es computacionalmente costoso (Dourado and Flores 2014).

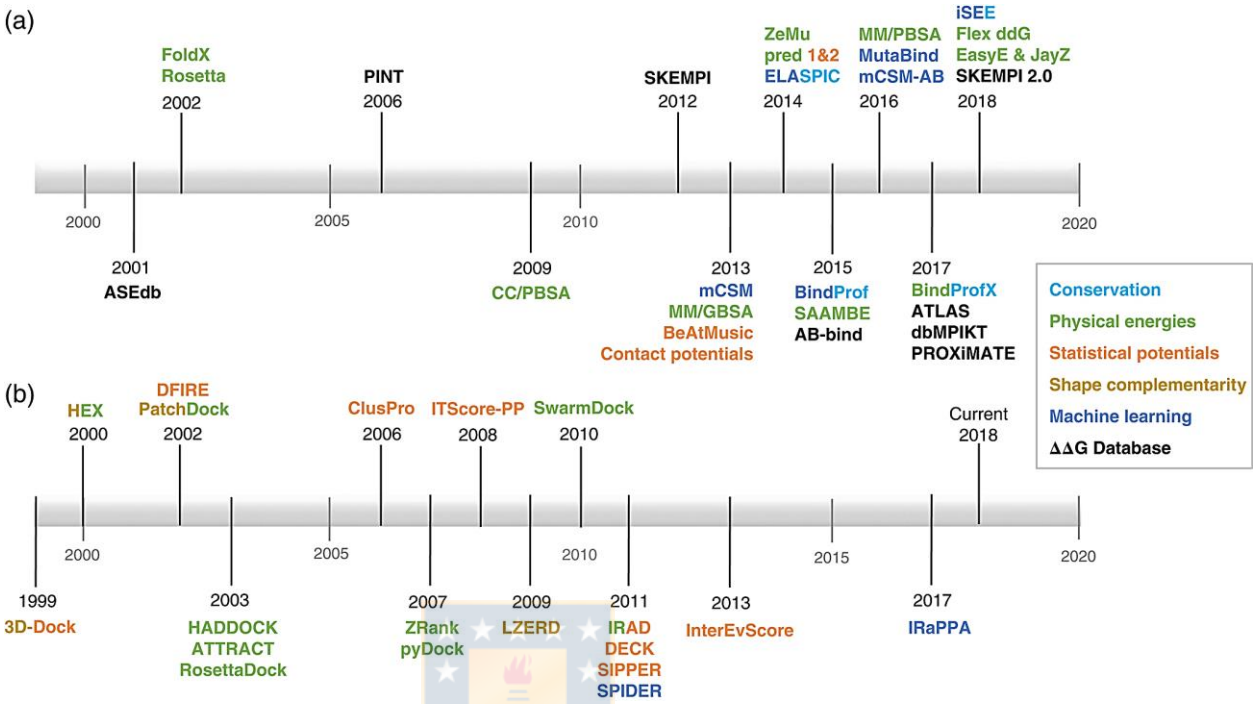

Figura 2. Línea de tiempo con predictores, bases de datos y funciones de puntaje representativas.
(a) predictores y bases de datos, en (b) funciones de puntaje **(Geng, Xue, et al. 2019)**.

Los potenciales estadísticos también se utilizan para estimar el ΔΔG. Estos son típicamente construidos desde estructuras experimentales y por lo tanto son altamente dependientes de los datos de entrenamiento. Los autores de la base de datos SKEMPI (Moal and Fernández-Recio 2012) utilizaron los datos disponibles para desarrollar un predictor ΔΔG basado en potenciales estadísticos, los cuales fueron derivados a partir de contactos a nivel atómico y de residuos (Moal and Fernandez-Recio 2013). Los potenciales a nivel de residuo resultaron levemente superiores a los de nivel atómico, logrando un PCC de 0.73 (*Pearson's correlation coefficient o PCC*) sobre 1949 mutantes obtenidas desde

SKEMPI. Otro predictor basado en potenciales estadísticos es BeAtMuSiC (Dehouck et al. 2013), el cual toma una aproximación *coarse-grained* para construir el modelo. Reportaron un PCC de 0.68 y un RMSE (*root mean square error*) de 1.19 kcal/mol en mutantes simples obtenidas de SKEMPI 1.1.

Tomar en consideración los cambios conformacionales inducidos por una mutación es esencial para los predictores ΔΔG basados en energía, ya que las energías calculadas son sensibles a los detalles en la estructura utilizada. Los primeros predictores de ΔΔG como FoldX y Rosetta modelan los cambios en la conformación de las cadenas laterales utilizando una biblioteca de rotámeros. El principal defecto de esta metodología es la incapacidad de encontrar la conformación de menor energía en la mayoría de los casos, debido a efectos de borde al discretizar las conformaciones (Harder et al. 2010). No fue hasta 2018, donde predictores como EasyE lograron identificar mínimos globales energéticos con alta confianza usando el algoritmo *Cost Function Network* (Viricel et al., 2018), resultando en un mayor rendimiento comparado a FoldX. Sin embargo, esta aproximación no considera flexibilidad en la cadena principal. Desde los primeros predictores de ΔΔG, se han realizado muchos intentos por explorar la flexibilidad estructural para mejorar el rendimiento de los predictores. ZEMu (Dourado and Flores 2014) fue uno de los primeros en abordar este aspecto, combinando flexibilidad estructural junto a cálculos de ΔΔG con FoldX. El método limita el muestreo conformacional en una pequeña región alrededor del sitio de la mutación y mantiene las regiones distantes sin perturbaciones, bajo la premisa

de que las mutaciones no inducen cambios globales en la estructura terciaria (Shanthirabalan, Chomilier, and Carpentier 2018). Esto llevó a una mejora en el PCC de 0.49 para FoldX a 0.62 para ZEMu sobre 1254 mutantes simples y múltiples de SKEMPI. Flex ΔΔG es otro predictor de ΔΔG reciente, el cual modela los cambios conformacionales usando la función energética denominada Talaris de Rosetta (Song et al. 2011). A diferencia de ZEMu, que solo limita el muestreo a una pequeña región alrededor del sitio mutado, Flex ΔΔG considera flexible toda la estructura generando un ensamble de conformaciones con el protocolo "backrub" de Rosetta (Smith and Kortemme 2008), aplicando minimización del ángulo de torsión y *repacking* de cadenas laterales. El ΔΔG es estimado a partir del promedio de las energías calculadas con Rosetta full-atom a través de todo el ensamble. En la publicación de Flex ΔΔG reportaron una mejora en las predicciones comparado a ZEMu en mutaciones tales como: mutaciones "small-to-large" (PCC de 0.48 vs 0.65), mutaciones múltiples "nonalanine" (PCC de 0.55 vs 0.63), y mutaciones en la interfaz antígeno-anticuerpo (PCC 0.54 vs 0.61).

Otro grupo importante de predictores clásicos son los basados en métodos de *Molecular Mechanics*/*Poisson-Bolztmann surface area (MM/PBSA)* o *Molecular Mechanics*/*Generalized Born surface area (MM/GBSA).* Estos métodos principalmente utilizan ensambles conformacionales desde simulaciones de dinámica molecular (Dehouck et al. 2013). La base de datos SKEMPI 2.0 (Jankauskaite et al. 2019) provee un incremento de datos de un 133%, lo que ha sido aprovechado para desarrollar nuevas herramientas de predicción.

1.6 Maduración *in silico* de la afinidad de anticuerpos

Los anticuerpos pueden ser optimizados con o sin antígenos. En la ausencia de antígeno los métodos computacionales ayudan a diseñar aspectos como la estabilidad, optimizando la secuencia de aminoácidos de la estructura del anticuerpo inicial. Con la estructura de un antígeno, se pueden considerar las interacciones antígeno-anticuerpo y optimizar la interfaz, con el objetivo de aumentar la afinidad del anticuerpo hacia su antígeno. Ambas aproximaciones de aumento en la afinidad y estabilidad son cruciales en el diseño de anticuerpos y el desarrollo de biofármacos.

La maduración natural de la afinidad de anticuerpos puede ser experimentalmente emulada a través de múltiples iteraciones de mutagénesis y selección, resultando en un anticuerpo con propiedades de unión deseadas. La identificación de mutaciones “favorables” para la maduración de la afinidad es un problema de combinatoria enorme (gran espacio muestral)(Barderas et al. 2008). En general, podemos encontrar aprox 60 residuos a través de los 6 CDRs. Si consideramos que cada posición puede mutar hacia cualquiera de los 20 aminoácidos estándares, nos da un total de 20^{60} combinaciones de anticuerpos. Por lo tanto, se hace necesario el uso de métodos computacionales de alto rendimiento capaces de explorar el espacio de búsqueda eficientemente.

El esquema más simple de maduración *in silico* de la afinidad requeriría primero la estructura del complejo antígeno-anticuerpo, ya sea experimental o un modelo.

Posteriormente, el esquema debería considerar el análisis de la afinidad de unión del complejo, para luego predecir cómo las mutaciones cambian la afinidad del anticuerpo. Una vez que las mutantes favorables son identificadas, las mutaciones pueden ser validadas experimentalmente.

Aún existen pocos estudios que sigan este tipo de aproximación para la maduración *in silico* de la afinidad de anticuerpos. Clark et al. 2006 realizó la maduración del anticuerpo AQC2 anti-VLA1 logrando un incremento en la afinidad 10 veces superior. Además, cristalizaron el anticuerpo mutante para ilustrar el impacto estructural de las 4 mutaciones sugeridas. Lippow, Wittrup, and Tidor 2007, validaron experimentalmente 2 de sus casos de estudios, demostrando un incremento de 140 veces en afinidad para el anticuerpo D44.1 anti-lisozima, y 10 veces para cetuximab. Su método fue computacionalmente costoso; testear 1080 mutantes simples del anticuerpo D44.1 requirió 24 horas en un clúster de 100-CPU. Kiyoshi et al. 2014 usaron herramientas disponibles en softwares como MOE y Discovery Studio para aumentar 4.6 veces la afinidad del anticuerpo 11K2. Ellos propusieron 12 mutaciones para incrementar la afinidad, aunque la validación experimental mostró que sólo 5 de estas mutaciones fueron beneficiosas. El problema central de estos estudios fue la habilidad para predecir la afinidad del anticuerpo. Por ejemplo, para una de las mutantes simples del anticuerpo D44.1 de Lippow, Wittrup, and Tidor 2007, se calculó un $\Delta\Delta G$ de -0.97kcal·mol^{-1}, sin embargo, $\Delta\Delta G$ experimental fue de 0.45kcal·mol^{-1}. La diferencia de 1.42kcal·mol^{-1} entre el valor calculado y el

experimental representa un error de 11 veces en la estimación de los valores K_D. De manera similar, Kiyoshi et al. 2014 predijo para una mutación de un residuo Asn en el loop CDR-L1 un ΔΔG de -15.4kcal·mol^{-1}. Sin embargo, el ΔΔG experimental fue solo de -1.0kcal·mol^{-1}, lo cual representa un error de 10^9 veces en la estimación de las K_D.

1.7 Fundamentos Metodológicos

Este trabajo se desarrolló en el ámbito de la bioinformática, y una de las principales motivaciones fue abordar desde un enfoque sinérgico el uso de métodos de aprendizaje automático.

1.7.1 Aprendizaje automático en bioinformática

La creciente cantidad de datos biológicos disponibles plantea dos desafíos: 1) el correcto almacenamiento y manejo de la información, y 2) la extracción de la información a partir de los datos. El segundo problema es uno de los desafíos claves en bioinformática, lo cual requiere el desarrollo de herramientas y métodos capaces de transformar todos estos datos en conocimiento biológico. Estas herramientas y métodos deben permitirnos ir más allá de una simple descripción de los datos y proveernos de conocimiento en forma de modelos comprobables. Es aquí donde el aprendizaje automático (también conocido en inglés como

machine learning, ML), con una gran variedad de técnicas computacionales, es capaz de resolver problemas biológicos complejos.

El aprendizaje automático es un proceso adaptativo que permite a los computadores aprender a partir de experiencias, aprender mediante ejemplos y aprender mediante analogía. Se han planteado una serie de razones por la cual el aprendizaje automático es ampliamente utilizado en bioinformática (Prompramote et al., 2005):

- **Permite explicar las reglas que se aplican en la práctica**. Los expertos no siempre son capaces de describir los factores que tomaron en cuenta al evaluar una situación. El aprendizaje automático puede servir de ayuda para extraer la descripción oculta en la situación, en términos de esos factores, y luego proponer las reglas que mejor describen el comportamiento.
- **Describe las propiedades o características que están ocultas en la complejidad de los datos**. Debido a la inherente complejidad de los organismos biológicos, los expertos están comúnmente confrontados a resultados no deseados. La causa de esto puede ser la presencia de propiedades desconocidas. El aprendizaje automático puede proveer pistas para posteriormente describir las propiedades o características que están ocultas para el experto.
- **Es adaptable y flexible**. Debido a que nuevos datos y nuevos conceptos son generados cada día en el área de la biología molecular, es esencial

aplicar técnicas capaces de ajustarse a estos rápidos cambios. El aprendizaje automático puede ser eficientemente adaptado a ambientes dinámicos.

- **Disminuye el efecto del ruido**. El aprendizaje automático es capaz de lidiar con la abundancia de ruido y/o ausencia de datos desde distintos escenarios biológicos.
- **Trabaja con *big data***. El aprendizaje automático es capaz de lidiar con un gran volumen de datos generados por los nuevos dispositivos de alto rendimiento, en orden de extraer relaciones ocultas existentes y que no son evidentes para los expertos.
- **Permite explorar relaciones durante el procesamiento del fenómeno**. En muchos escenarios biológicos, los expertos sólo pueden especificar pares de datos de entrada y salida, y no son capaces de describir la relación general entre las diferentes características que podrían servir para describir cómo se relacionan. El aprendizaje automático es capaz de ajustar su estructura interna a los datos existentes, produciendo modelos y resultados aproximados.

En el área de investigación bioinformática, los métodos de aprendizaje automático son utilizados, por ejemplo, en el descubrimiento de nuevos fármacos (Zernov et al., 2003), analizar y predecir enfermedades (Cruz and Wishart, 2007), clasificar elementos genéticos similares (Brown et al., 2000), predecir localización

subcelular de proteínas (Chou and Shen, 2010), predecir el plegamiento de proteínas (Wei and Zou, 2016), predecir interacción proteína-proteína (Saha et al., 2014), entre otros. La figura 3 muestra un esquema general de las aplicaciones actuales del aprendizaje automático en bioinformática.

Dependiendo del objetivo del estudio y las características de los datos disponibles, los métodos de aprendizaje automático pueden ser divididos en:

- **Aprendizaje supervisado**. A partir de un grupo de datos de entrenamiento, que consiste en casos de entrada y salidas deseadas, el algoritmo genera una función (o modelo) para predecir la salida de futuros casos cuya salida es desconocida. Cuando la salida del objeto es una variable continua (por ejemplo, altura, precio del dólar), el problema es conocido como regresión. Cuando la salida (o etiqueta) es una categoría (o valor discreto), el problema es conocido como clasificación.
- **Aprendizaje no supervisado**. A partir de un grupo de datos de entrenamiento, que consisten solo en casos de entradas, el algoritmo es capaz de agrupar los objetos de manera tal que los objetos del mismo grupo, también denominado clúster, son más similares entre sí que aquellos objetos en otros grupos. En contraste al aprendizaje supervisado, no existen etiquetas en los datos.

Entre el aprendizaje supervisado y el no supervisado, existe el aprendizaje semi-supervisado. También existe el aprendizaje reforzado, en el cual el programa

computacional interactúa con un ambiente dinámico (por ejemplo, conducir un auto).

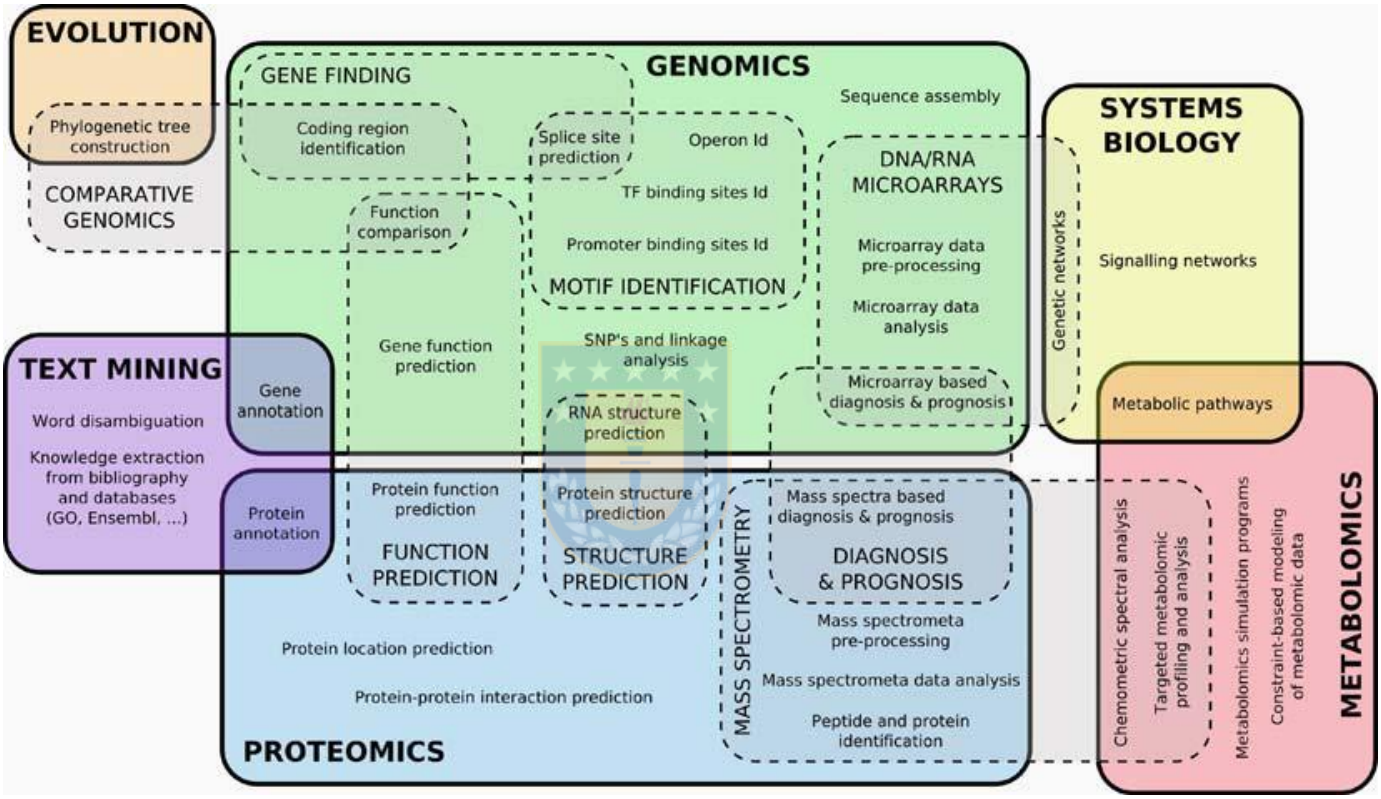

Figura 3. Aplicaciones actuales del aprendizaje automático en bioinformática.
Diferentes áreas de la biología en donde algunos problemas pueden ser resueltos mediante aprendizaje automático. Las áreas se clasifican en Evolución, Genómica, Biología de Sistemas, Minería de texto, Proteómica y Metabolómica. (Inza et al., 2010)

1.7.2 Aprendizaje Automático

El flujo de trabajo general de las aplicaciones de aprendizaje automático es descrito en 4 pasos: 1) Obtención y representación de los datos, 2) pre-procesamiento de los datos, 3) entrenamiento del modelo, y 4) evaluación del modelo (Figura 4). Se debe tener en cuenta que no siempre se sigue este esquema, y dependerá mucho del problema y el tipo de datos iniciales.

1.7.2.1 Obtención y representación de los datos

El primer paso antes de modelar un problema, es la obtención de un conjunto de objetos de entrenamiento representativos del problema en cuestión. Muchas veces es necesario seleccionar, limpiar y convertir los datos iniciales, de manera que tengan un formato adecuado para la aplicación de algoritmos de aprendizaje automático. La forma más común de representar los datos es de forma tabular, una tabla de datos donde cada fila es una muestra u objeto, y cada columna es una variable.

La representación de los datos tiene como finalidad encontrar aquellas características que representen de mejor manera a cada tipo de objeto y su elección es altamente específica al problema. Es habitual denotar una muestra de datos u objeto, incluyendo todas las co-variables y características como una

entrada X (usualmente un vector de números), y etiquetarla con su variable respuesta o valor de *salida Y* (usualmente un único número), si está disponible.

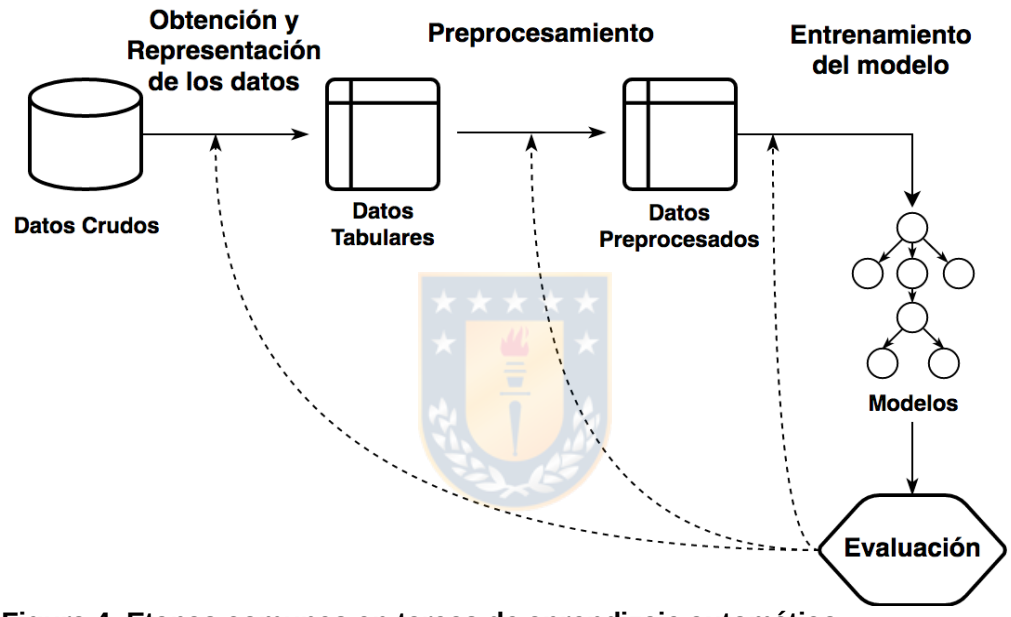

Figura 4. Etapas comunes en tareas de aprendizaje automático.
En general el desarrollo de modelos de aprendizaje automático involucra las siguientes etapas: 1) Obtención y representación de los datos, 2) pre-procesamiento de los datos, 3) entrenamiento del modelo, y 4) evaluación del modelo.

1.7.2.2 Preprocesamiento

Previo a cualquier aplicación directa de los algoritmos de aprendizaje automático, es esencial siempre revisar la calidad de los datos, y realizar un paso de

preprocesamiento con el objetivo de refinar/depurar los datos, siendo esta una de las tareas que mayor tiempo consumen. Las tareas comúnmente realizadas en esta etapa son:

- **Formatear**: Los datos seleccionados puede que no estén en el formato indicado para trabajar. Como se mencionó anteriormente, una de las formas más adecuadas de almacenar los datos es de forma tabular.
- **Limpiar**: Limpiar los datos es remover o agregar datos faltantes. Puede haber muestras en los datos que estén incompletos. Estas muestras podrían necesitar ser removidas. Adicionalmente, puede haber información sensible en los datos que requiera ser puesta bajo el anonimato o removida completamente.
- **Muestrear**: Pueden existir muchos más datos disponibles de los necesarios para trabajar, lo que puede resultar computacionalmente costoso. Por otro lado, en muchos casos, cuando la cantidad de datos no es suficiente para tener un conjunto de prueba independiente, se opta por muestrear directamente desde los datos iniciales.

Uno de los últimos pasos de preprocesamiento es comúnmente llamado transformación de los datos. Dentro del conjunto de características que representan a cada objeto, pueden presentarse casos en que los rangos de valores sean muy variados, o que las escalas sean diferentes, lo cual puede influenciar de forma directa en el algoritmo de aprendizaje automático a utilizar. Algunos de las transformaciones usadas son:

- **Normalización z-score:** Varios algoritmos de aprendizaje automático requieren que los datos posean las propiedades de una distribución normal estándar, es decir media igual a cero (μ= 0) y desviación estándar igual a uno (σ = 1).
- **Normalización Min-Max:** Otra alternativa es transformar los datos para que estén situados entre un valor mínimo y máximo deseado.

1.7.2.3 Entrenamiento de un modelo

Cuando hablamos de entrenar un modelo, nos referimos al proceso que involucra proveer a un algoritmo de aprendizaje automático con **datos de entrenamiento**, con el fin de ajustar una función de mapeo inducida directamente de patrones presentes en los datos.

Luego de entrenar un modelo de aprendizaje automático con los datos de entrenamiento, es necesario evaluar el modelo con otros datos, denominado a estos de **datos de prueba**. El propósito de esta etapa es mostrar que el modelo no ha aprendido simplemente a recordar los datos de entrenamiento, sino que realmente aprendió patrones significativos que generalizan con observaciones fuera de los datos de entrenamiento.

1.7.2.4 Evaluación y selección del modelo

Comúnmente, el aprendizaje automático implica mucha experimentación, por ejemplo, afinar los botones internos de un algoritmo de aprendizaje, llamados hiper-parámetros. Si ejecutamos un algoritmo de aprendizaje sobre un conjunto de datos de entrenamiento con diferentes híper-parámetros obtendremos diferentes modelos. Ya que queremos seleccionar el modelo con mejor rendimiento de todos, es necesario estimar los rendimientos respectivos para compararlos. Además, de simplemente ajustar el algoritmo, usualmente no se experimenta con un solo algoritmo, sino que lo ideal es comparar diferentes algoritmos unos con otros. Por lo tanto, el rendimiento predictivo de los modelos se evalúa para:

1. Estimar el rendimiento generalizado, esto quiere decir el rendimiento predictivo del modelo sobre datos futuros (desconocidos).
2. Aumentar el rendimiento predictivo ajustando los híper parámetros de un algoritmo en particular, y seleccionando el mejor modelo entregando diferentes configuraciones de híper parámetros.
3. Identificar el algoritmo de aprendizaje automático más indicado para nuestro problema comparando diferentes algoritmos y seleccionando aquel con mayor rendimiento, además se considera el punto 2 para cada algoritmo en particular.

Una de las herramientas que permite visualizar el desempeño de un algoritmo en tareas de clasificación es la matriz de confusión (Ting, 2011). Esta corresponde a una matriz de 2 dimensiones, en la cual cada columna representa el número de predicciones de cada clase mientras que cada fila representa las instancias reales de cada clase. La matriz de confusión tiene como beneficio la capacidad de ver si el modelo está confundiendo dos clases.

En la Tabla 1 observamos un problema de clasificación de 2 clases (positivo y negativo). A partir de la tabla se pueden calcular algunas métricas básicas para la evaluación del modelo, considerando los valores *a*, *b*, *c* y *d* se tiene:

- Verdaderos positivos o sensibilidad (*recall*), la porción de muestras positivas que el modelo predice correctamente: $a/(a+b)a/(a+b)$
- Verdaderos negativos o especificidad, la porción de muestras negativas que el modelo predice correctamente: $d/(c+d)d/(c+d)$
- Precisión (*precision*), la proporción de resultados positivos que son verdaderos positivos: $a/(a+c)a/(a+c)$
- Exactitud (*accuracy*), la proporción total de predicciones que fueron correctas: $a+d/(a+b+c+d)a+d/(a+b+c+d)$
- Valor-F (F-score, F1), puede ser interpretado como un promedio ponderado de la precisión y la sensibilidad:

  $F1 = 2 * (precision * sensibilidad)/(precision + sensibilidad)$

Tabla 1. Matriz de confusión para un problema de 2 clases.

		Clase Predicha	
		Positivo	**Negativo**
Clase real	**Positivo**	a	b
	Negativo	c	d

En tanto para evaluar problemas de regresión, la mayoría de ellos se basan en la similitud de los valores predichos y reales. Una de las métricas más simples para calcular la precisión del modelo es el error calculado como la diferencia promedio entre los valores predichos y los reales para todas las filas. El error de un modelo de regresión es la diferencia entre los datos puntuales y la línea de tendencia generada por el algoritmo. Existen distintas maneras de calcular el error pero las más comunes son:

- **Error medio absoluto (MAE)** es la media del valor absoluto de los errores. Es la medida más fácil de entender, ya que es sólo el error promedio.
- **Error cuadrático medio (MSE)** es la media de los errores al cuadrado. Es más popular que el error medio absoluto porque el enfoque se orienta más hacia grandes errores. Esto se debe a que el término al cuadrado aumenta exponencialmente los errores más grandes en comparación con los más pequeños.
- **Raíz del error cuadrático medio (RMSE)** es la raíz cuadrada de la anterior medida. Esta es una de las métricas más utilizadas porque es

interpretable en las mismas unidades que el vector de respuesta haciendo fácil de correlacionar con la información.

Todas estas métricas pueden ser comparadas al momento de evaluar el rendimiento del modelo. El esquema más sencillo para estimar el rendimiento predictivo es entrenar el modelo sobre la totalidad de los datos y evaluarlo sobre los mismos. Sin embargo, esta es una de las peores metodologías de evaluación (llamado evaluación de re-substitución), ya que introduce un sesgo optimista elevado debido al sobreajuste.

Uno de los esquemas más utilizados para la evaluación y selección de modelos es la técnica de validación cruzada de K iteraciones (k-*fold cross-validation*). La idea principal de la técnica es que cada muestra de datos tiene la oportunidad de ser evaluada. Se itera sobre los datos K veces, y en cada ronda se dividen los datos en K partes: una parte es usada para la evaluación, y el resto de las partes (K-1) son unidas como conjunto de entrenamiento para la evaluación del modelo. La Figura 5 ilustra el proceso de una validación cruzada de 5 iteraciones.

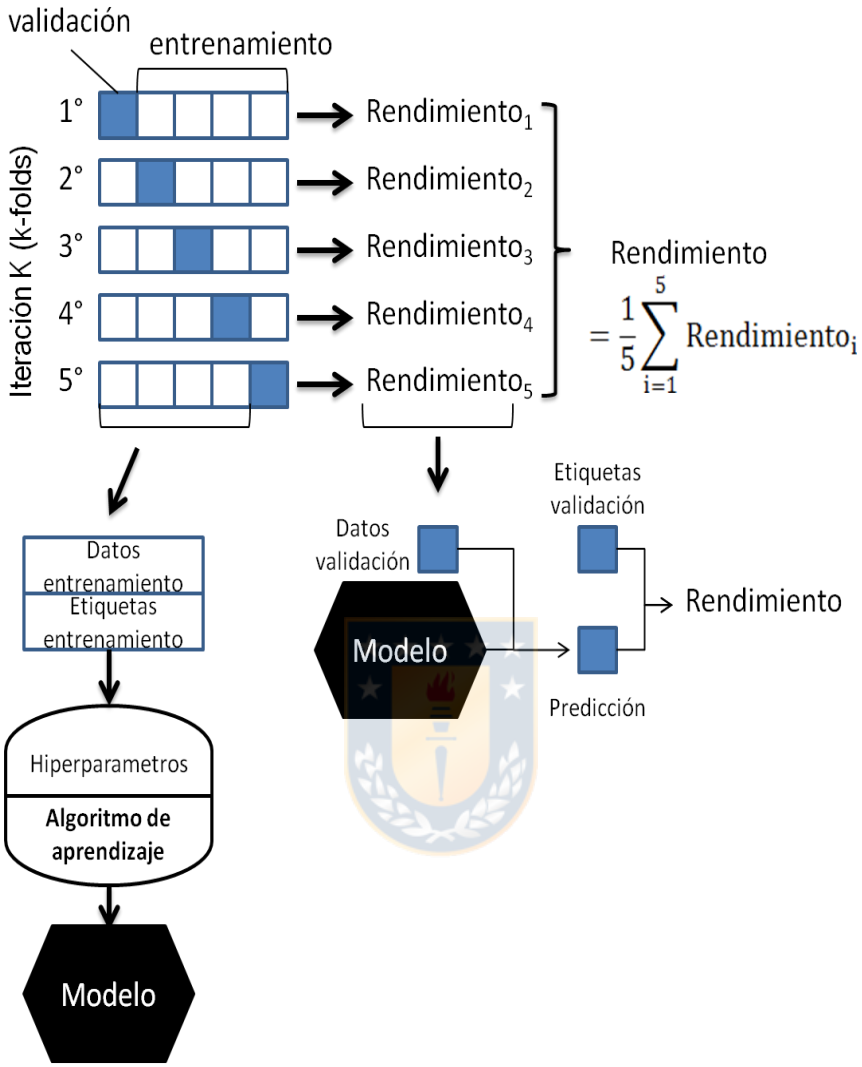

Figura 5. Evaluación de un modelo utilizando validación cruzada de 5 iteraciones.
En este ejemplo, el esquema de validación cruzada se basa en particionar los datos en 5 partes, entrenar el modelo con 4/5 de los datos y se valida con 1/5. Se repite el proceso 5 veces, en cada iteración se rota el grupo de validación. El rendimiento final se obtiene promediando cada iteración.

1.7.3 Análisis exploratorio de datos

El análisis exploratorio de datos (EDA) es un paso esencial en cualquier análisis de investigación. El objetivo principal es examinar la distribución, ruido y anomalías de los datos, guiando el proceso evaluativo de la hipótesis (Tukey, 1977). También provee herramientas para la generación de la hipótesis mediante la visualización y comprensión de los datos, usualmente a través de la representación gráfica. El análisis exploratorio de datos asiste en el reconocimiento de patrones ocultos, identificar si existe alguna estructura particular, y diferenciar entre puntos de interés o ruido. Por otra parte, también se puede utilizar para preparar los datos previa aplicación de métodos de aprendizaje automático. Todo lo anterior se puede lograr utilizando diferentes técnicas estadísticas, pero que no reemplazan a los métodos de visualización. En particular, algunas de las técnicas gráficas que se utilizan en EDA son: Gráfico de dispersión, histograma, gráfico de cajas, reducción dimensional, análisis de agrupamiento (*clustering*), entre otros.

1.8 Planteamiento

Los anticuerpos son una de las moléculas más importantes en la investigación de biofármacos. Actualmente, conocemos 78 anticuerpos monoclonales aprobados por la FDA, y muchos otros se encuentran en las últimas etapas clínicas. Esto ha impulsado el desarrollo de técnicas experimentales para diseñar anticuerpos (Ab) de alta especificidad y afinidad por su antígeno. Sin embargo, estos métodos poseen sus limitantes, y cada vez es más claro que los métodos de diseño computacional de anticuerpos facilitan este proceso

La maduración por afinidad de anticuerpos es un proceso natural, y se puede imitar experimentalmente en el laboratorio a partir de bibliotecas de diversidad y selección de epítopos. Sin embargo, sigue siendo dificultoso apuntar a epítopes específicos solo con esta tecnología, por lo cual una ruta alternativa es combinarlos junto al diseño computacional. La secuenciación de nueva generación, el modelamiento tridimensional y el docking molecular han permitido realizar diseño racional de anticuerpos sin la necesidad del complejo cristalográfico (Ambrosetti et al. 2019).

Este proceso es asistido por computadora y algoritmos complejos basados en las propiedades biofísicas de las fuerzas intermoleculares que impulsan la energía de unión. A pesar de la diversidad de estos métodos, aún existe la necesidad de desarrollar predictores fiables y rápidos que permitan estimar los

efectos de mutaciones en la afinidad de unión (es decir, el cambio de energía libre de unión entre un complejo nativo y mutante, ΔΔG). Métodos computacionales de tales características pueden ser de gran ayuda en procesos como la maduración *in silico* de la afinidad de anticuerpos, donde mediante mutagénesis saturante, se puede generar combinatorias de los 20 aminoácidos naturales en más de 8 posiciones llegando al orden de 10^{12} (Hu et al. 2015).

La creciente cantidad de secuencias e información estructural ha centrado el interés en la aplicación de aprendizaje automático para el desarrollo de herramientas predictivas en múltiples áreas de la bioinformática. En tareas como la predicción de afinidad de anticuerpos han demostrado superar a métodos clásicos, como las funciones empíricas o las basadas en conocimiento (Rodrigues et al. 2019). Sin embargo, existen pocos métodos computacionales que aborden directamente el problema de la maduración *in silico* de la afinidad de anticuerpos, de tal manera que permita hacer menos costoso computacionalmente este proceso.

En este trabajo se propone el desarrollo de un nuevo método basado en aprendizaje automático específico para complejos Antígeno-Anticuerpo capaz de predecir el cambio de afinidad por mutaciones puntuales, llamado ***ABPRED***. Para ello se obtuvo una base de datos de constantes cinéticas para mutantes simples en complejos Antígeno-Anticuerpo. Se modelaron las mutantes y retromutantes, para posteriormente calcular características estructurales y energéticas con las cuales entrenar y evaluar un modelo. Finalmente se comparó ABPRED frente a

otros predictores representativos actualmente disponibles como FoldX, BeatMusic, DFIRE2 y mCSM-AB.

1.8.1 Hipótesis

Un predictor basado en aprendizaje automático específico para complejos antígeno-anticuerpo, puede estimar el cambio de afinidad de mutaciones puntuales con precisión y correlación comparables a métodos clásicos.

1.8.2 Objetivo general

Desarrollar un método para la predicción del cambio de afinidad por mutaciones puntuales en complejos antígeno-anticuerpo.

1.8.3 Objetivos Específicos

1.- Obtener una base de datos de constantes cinéticas experimentalmente determinadas para mutaciones puntuales en complejos Antígeno-Anticuerpo.

2.- Generar y evaluar un conjunto de características que modelen diferentes aspectos involucrados en el reconocimiento antígeno-anticuerpo y los efectos en sus mutaciones.

3.- Construir un modelo de aprendizaje automático utilizando las características calculadas y los datos experimentales de las mutantes.

4.- Comparar el rendimiento del modelo predictivo contra otros predictores de cambios de energía libre del estado del arte.

2 MATERIALES Y MÉTODOS

En este capítulo se presenta la metodología desarrollada, la cual se compone de 3 etapas principales. La Figura 6 presenta un flujo general de la metodología:

1) **Generación del conjunto de datos.** Esto comprende a) la obtención de una base de datos de constantes cinéticas experimentalmente determinadas para mutaciones puntuales en complejos Ag-Ab, y b) la extracción de características.
2) **Construcción modelo predictivo.** Comprende el entrenamiento y evaluación del modelo de aprendizaje automático llamado ***ABPRED***, utilizando una metodología de *stacking*.
3) **Comparación de rendimiento.** Tiene como objetivo comparar ***ABPRED*** con otros predictores de ΔΔG representativos mediante una prueba de simple ciego con datos experimentales obtenidos de ADAPT.

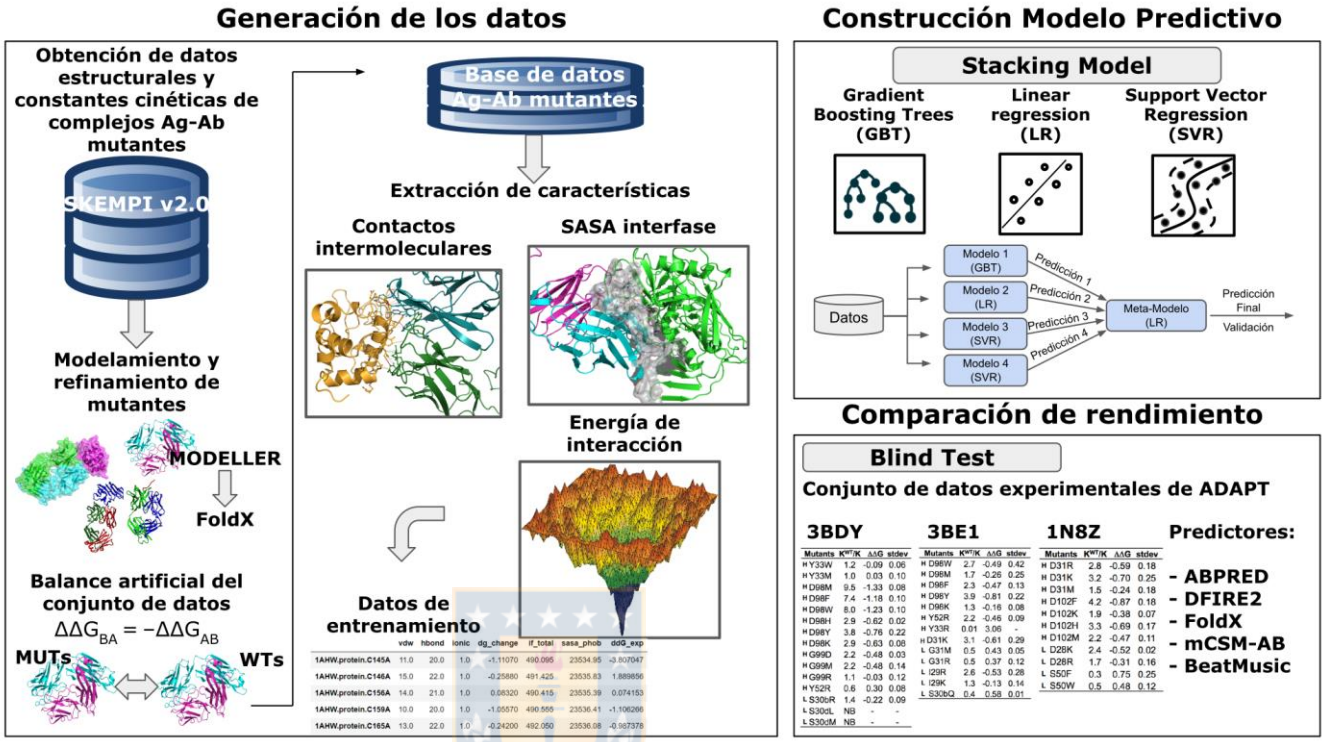

Figura 6. Resumen del flujo de trabajo
En la figura se observa tres paneles: izquierda) Obtención de los datos estructurales de los complejos y mutantes a incluir para el desarrollo de modelos, la información de mutantes reversas es incorporada por medio de modelamiento molecular con Modeller y refinada la interfaz con FoldX; luego en el panel central) se observa la obtención de características moleculares desde los complejos de la base de datos de antígenos-anticuerpos previamente generada; finalmente, en el panel de la derecha se desarrolla un modelo desde los datos usando metodología stacking y distintos algoritmos, para finalmente aplicar una prueba ciega con complejos desde datos experimentales y comparar con métodos desde el estado del arte.

2.1 Obtención y procesamiento de los datos

Para desarrollar ***ABPRED*** se recolectaron datos de afinidades de unión con estructuras experimentalmente determinadas desde las bases de datos AB-BIND y SKEMPI2.0, con los cuales se entrenó y probó ***ABPRED***. La base de datos AB-

BIND es una colección de parámetros termodinámicos de complejos antígeno-anticuerpo mutantes y nativas, incluyendo el ΔΔG de unión, asociado a estructuras cristalográficas de los complejos. Desde AB-BIND se consideró una primera selección de sólo **mutantes simples**, posteriormente se descartaron mutantes de modelos comparativos y se filtraron mutantes redundantes.

En cuanto a la base de datos SKEMPI2.0, esta contiene datos sobre los cambios en parámetros termodinámicos y constantes cinéticas debido a mutaciones puntuales en una gran variedad de complejos estructurales. En esta base de datos una primera selección consistió en filtrar por **mutantes simples** y sólo datos para complejos Ag-Ab. Esto se realizó usando la información definida en `Hold_out_type` seleccionando `AB/AG` y `AB/AG,Pr/PI`. Varias de las mutantes en SKEMPI2.0 tienen más de una afinidad de unión medida experimentalmente, por lo que se consideró sólo aquellas cuyos métodos eran `SPR`, `KinExA` o `IASP`. Finalmente se filtraron mutantes no-ligantes (non-binders).

Con el fin de generar una base de datos mayor con la cual entrenar ***ABPRED***, se combinaron los datos de mutantes obtenidos desde AB-BIND y SKEMPI2.0, generando la base de datos **ABPRED_DB**. Para esto se realizó un análisis de conjuntos entre los datos de AB-BIND y SKEMPI2.0, con el fin de evitar mutantes redundantes. El análisis consideró el identificador (ID) de las cadenas, ya que para un mismo complejo el ID de sus cadenas puede diferir entre ambas bases de datos. Por ejemplo, para la estructura cristalográfica 1VFB las cadenas liviana

y pesada del anticuerpo están identificadas como A y B en SKEMPI2.0, y como H y L en AB-BIND. Considerando lo anterior, para aquellas estructuras cuyo ID de cadena era diferente entre ambas base de datos, se optó por mapear los ID de la estructura en AB-BIND a su correspondiente ID en SKEMPI2.0.

2.2 Balance de datos y modelamiento de mutantes

Dado que la formulación de energía libre de Gibbs es una función de estado termodinámico, un cambio en la afinidad de unión de una mutación desde una proteína nativa a su mutante ($\Delta\Delta G_{WT \rightarrow MT}$) debería ser equivalente al cambio negativo de energía libre de unión de la hipotética mutación inversa (retromutante), desde la mutante a la proteína nativa ($-\Delta\Delta G_{MT \rightarrow WT}$) (Thiltgen and Goldstein 2012). La mayoría de las mutantes en la base de datos AB-BIND y SKEMPI2.0 son desestabilizantes ($\Delta\Delta G > 0$), lo cual puede significar un problema a la hora de construir modelos predictivos debido al imbalance de datos. Es por esto que al momento de modelar las mutantes obtenidas en **ABPRED_DB** también se consideraron las hipotéticas mutaciones inversas, como se ha propuesto anteriormente. Todas las mutantes fueron modeladas con MODELLER (Webb and Sali 2016) utilizando el protocolo de mutación puntual (https://salilab.org/modeller/wiki/Mutate%20model), modificado para un refinamiento de nivel rápido.

2.3 Extracción de características

Debido a la esencia basada en datos del aprendizaje automático, el acceso a una gran cantidad de datos experimentales y la construcción de características que reflejen los cambios estructurales y físico-químicos causados por mutaciones, son un factor crucial para el éxito de estos métodos.

En este trabajo se utilizaron 2 principales clases de características para entrenar ***ABPRED***: características estructurales (Interacciones no covalentes de la interfaz y área accesible a solvente, SASA) y basadas en energía (términos energéticos). Previo al cálculo de características, cada mutante modelada es reparada 2 veces consecutivas implementando la función *RepairPDB* de FoldX. Este paso prepara la estructura proteica disminuyendo artefactos, al identificar y reparar aquellos residuos que tienen malos ángulos de torsión y solapamientos de las superficies de Van der Waals. Adicionalmente, es un requisito previo para usar otras funciones de FoldX implementadas ya que *RepairPDB* utiliza el campo de fuerza específico de FoldX para esta tarea.

2.3.1 Interacciones no covalentes de la interfaz

Se realizó un análisis de las interacciones no covalentes en la interfaz antígeno-anticuerpo de las mutantes y retromutantes utilizando los contactos calculados por Arpeggio (Jubb et al. 2017). De los resultados se consideró la sumatoria de

los 15 tipos de interacción átomo-átomo descritos por Arpeggio: *clash, covalent, vdw_clash, vdw, proximal, hbond, weak_hbond, xbond, ionic, metal_complex, aromatic, hydrophobic, carbonyl, polar, weak_polar*. Desde aquí se definieron las siguientes características relevantes: **número de contactos en la interfaz mutante y la diferencia del número de contactos entre interfaz mutante y retromutante**.

2.3.2 SASA de interfaz

La accesibilidad al solvente o área superficial accesible al solvente (SASA), es considerado una de las principales características para determinar el plegamiento y estabilidad de proteínas (Syed et al., 2014). Esta se define por el área que dibuja el centro de una sonda a medida que se mueve sobre la superficie de la molécula. Para las proteínas, la sonda utilizada comúnmente como solvente es una representación esférica de una molécula de agua de radio 1.4 Å (Shrake and Rupley 1973). En este trabajo, los cálculos de SASA para cada complejo antígeno-anticuerpo fueron realizados con el programa POPS (Cavallo, Kleinjung, and Fraternali 2003). Desde aquí se definieron las siguientes características relevantes: **SASA total, hidrofílico e hidrofóbico del complejo**

mutante y el SASA total, hidrofílico e hidrofóbico de la interfaz mutante. El SASA de la interfaz (ISASA) es calculado como:

$$ISASA = (TA + TB) - T / 2$$

donde *T* es el SASA total del complejo, y *TA* y *TB* son el SASA total de antígeno y el anticuerpo respectivamente.

2.3.3 Términos energéticos

Se obtuvo información sobre la energía de interacción del complejo antígeno-anticuerpo implementando la función *AnalyseComplex* de FoldX. De esta forma se calcularon los términos energéticos para mutantes y retromutantes. Con esta información se definieron 2 set de características: la **diferencia de los términos energéticos entre estructura mutante y retromutante**, y los **términos energéticos de la estructura mutante**. En adición se incluye el **ΔΔG de FoldX** calculado usando la Ecuación 1.

En total se calcularon 77 características para cada estructura mutante y retromutante modelada. Estos datos dan forma al conjunto de datos iniciales ABPRED_DATA, que posteriormente es utilizado para entrenar y evaluar el ***ABPRED***.

2.4 Desarrollo de *ABPRED*

Este paso consiste en entrenar un modelo de aprendizaje automático utilizando la metodología de *stacking* o apilamiento de modelos. En toda esta etapa se utilizó la biblioteca de Python *scikit-learn* (Pedregosa et al. 2011). La figura 7 presenta un esquema de la metodología general utilizada para el desarrollo de ***ABPRED,*** la cual se puede dividir en 3 partes: en verde, pre-procesamiento y separación del conjunto inicial ABPRED_DATA, en morado, entrenamiento del modelo con metodología *stacking,* usando 4 algoritmos de regresión distintos, y por último en rojo, re-entrenamiento del mejor modelo con la totalidad de datos.

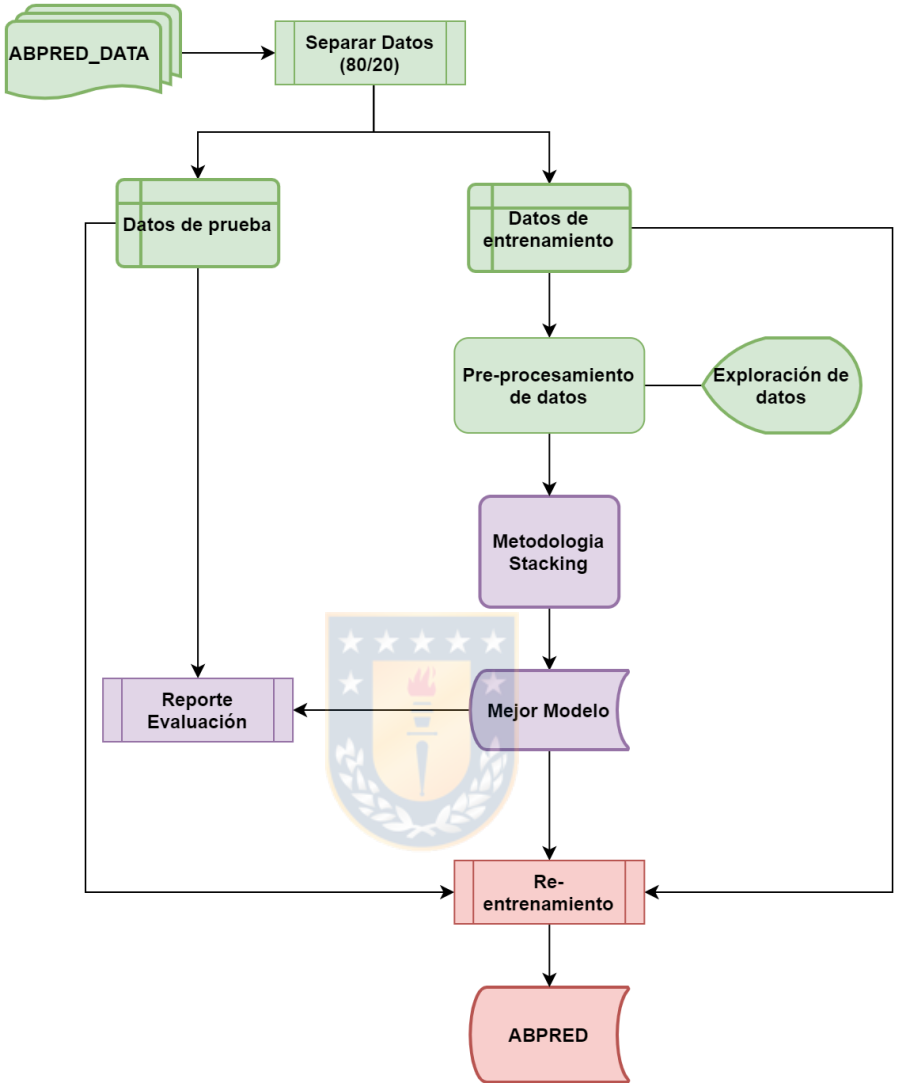

Figura 7. Metodología para el desarrollo de modelo ABPRED
En verde, separación del conjunto inicial ABPRED_DATA en datos de prueba y entrenamiento, y posterior pre-procesamiento de estos últimos. En morado, entrenamiento del modelo con metodología stacking. En rojo, re-entrenamiento del mejor modelo con la totalidad de datos.

2.4.1 Pre-procesamiento y separación de los datos

En esta etapa de pre-procesamiento, previa separación de datos, se realizó en primer lugar un filtro de valores ΔΔG anormales en ABPRED_DATA. Se consideró como no-ligantes todas aquellas mutantes con un ΔΔG igual a **-8 kcal/mol,** las que se escapan de la distribución esperada para este tipo de datos, además se ha reportado que impactan negativamente en la generación de modelos (Pires and Ascher 2016). Cabe mencionar que las equivalentes retromutantes también fueron filtradas.

Posteriormente se separó ABPRED_DATA en 80% datos de entrenamiento (ABPRED_DATA80) y 20% datos de prueba (ABPRED_DATA20), utilizando muestreo estratificado aleatorio. Como este es un problema de regresión y nuestra variable es continua, para poder conservar la distribución en ambos subconjuntos, se binarizo a partir de los valores de ΔΔG y posteriormente se aplicó el muestreo estratificado.

En este trabajo, el pre-procesamiento de ABPRED_DATA80 consistió principalmente en las siguientes tareas: explorar aquellas características que mejor correlacionan con la variable experimental ΔΔG e identificar anomalías a partir de las 3 mejores características que más correlacionan con la variable experimental.

Finalmente, los datos ABPRED_DATA80 son utilizados para entrenar un modelo de stacking y ABPRED_DATA20 es utilizado para una prueba

independiente con la cual obtener métricas de evaluación, como se explica a continuación.

2.4.2 Metodología de Stacking o Apilamiento

Para desarrollar ABPRED se utilizó la metodología de *stacking* o apilamiento (Wolpert 1992). El apilamiento es una técnica de ensamble utilizada para combinar información de múltiples modelos predictivos con el objetivo de generar un nuevo modelo. Proporciona un esquema para minimizar la tasa de error de generalización de uno o más modelos predictivos y se ha aplicado con éxito en diferentes tareas de aprendizaje automático (Nagi and Bhattacharyya 2013; Xiong et al. 2018:4; Yin et al. 2018; https://www.gormanalysis.com/blog/guide-to-model-stacking-i-e-meta-ensembling/).

La metodología de *stacking* implica dos etapas de aprendizaje y puede verse como una extensión del método de validación cruzada (Figura 8). Los modelos de la primera y segunda etapa se denominan modelos base y meta-modelos, respectivamente. En la primera etapa se emplea un grupo de modelos base. Luego, utilizando un meta-modelo en la segunda etapa, las predicciones de los modelos base se combinan con el objetivo de reducir el error de generalización. Es deseable utilizar modelos diferentes entre sí, particularmente diferentes

algoritmos, con el objetivo de enriquecer el meta-modelo con más información sobre el espacio de solución.

Para desarrollar ***ABPRED*** se utilizaron 4 modelos base diferentes en la primera etapa, usando 3 algoritmos: LASSO (Friedman, Hastie, and Tibshirani 2010), Gradient Boosted Regression Trees (GBRT o GBT)(Friedman 2002) y Support Vector Regression (SVR)(Smola and Schölkopf 2004). En la segunda etapa se utilizó como meta-modelo LASSO. Tanto la metodología stacking y los algoritmos descritos, fueron implementados usando las librerías de Python *scikit-learn* y *vecstack* (https://github.com/vecxoz/vecstack).

El modelo predictivo fue entrenado con ABPRED_DATA80 usando validación cruzada de 10 iteraciones, y posteriormente ABPRED_DATA20 se utilizó como prueba independiente con el fin de evaluar la capacidad de generalización del método.

2.4.3 Evaluación del Modelo

Para evaluar el rendimiento del modelo se calcularon el Coeficiente de Correlación de Pearson (PCC en inglés), la raíz del error cuadrático medio (RMSE en inglés) y el coeficiente de determinación R^2. La evaluación de rendimiento se consideró tanto en la validación cruzada y la prueba independiente.

2.5 Evaluación comparativa de ABPRED

Con el objetivo de comparar el rendimiento de ABPRED con otros predictores de ΔΔG representativos, se recolectó un conjunto de datos ciego, a partir del trabajo publicado por Vivcharuk et al. 2017. Este conjunto de datos llamado ADAPT36, consta de 36 mutantes no incluidas en ABPRED_DB, que comprenden 13 mutantes del complejo 3BDY, 13 mutantes del complejo 3BE1 y 11 mutantes para 1N8Z. En tanto los predictores seleccionados para la comparación fueron FoldX, BeatMusic, DFIRE2 y mCSM-AB.

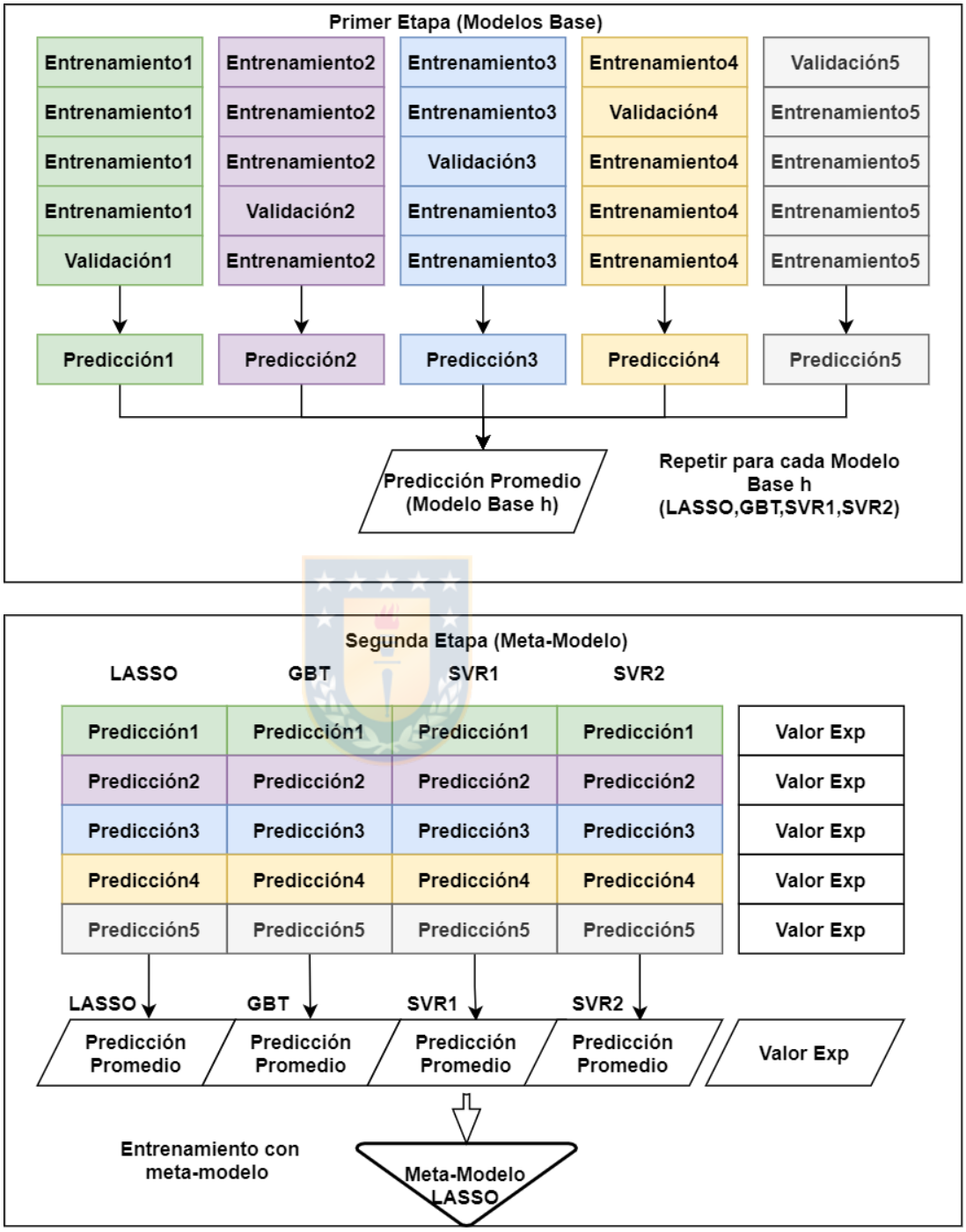

Figura 8. Metodología de stacking o apilamiento utilizada en

En la figura se observan 2 paneles: superior) En la primera etapa los modelos base (LASSO,GBT,SVR1,SVR2) son entrenados usando ABPRED_DATA80. inferior) En la segunda etapa, las predicciones de los modelos base son tomadas como input por el meta-modelo (LASSO). Figura adaptada de Xiong et al. 2018

4 RESULTADOS

El objetivo de esta investigación fue desarrollar un predictor del cambio de afinidad por mutaciones puntuales en complejos antígeno-anticuerpo. Para esto fueron necesarios realizar 4 objetivos específicos: Primero, se obtuvo una base de datos de constantes cinéticas experimentalmente determinadas para mutaciones puntuales en complejos Antígeno-Anticuerpo, llamada ABPRED_DB. Segundo, se generó un conjunto de características para modelar aspectos estructurales y energéticos de la interacción antígeno-anticuerpo y sus efectos en mutaciones. Tercero, se entrenó y evaluó ***ABPRED***, un predictor que utiliza la metodología *stacking* para predecir los cambios de afinidad. Por último, se comparó el rendimiento del modelo predictivo contra otros predictores representativos.

4.1 Base de datos ABPRED_DB

La selección inicial desde AB-BIND resultó en un total de 558 mutantes simples que comprenden 24 complejos Ag-Ab diferentes. Al filtrar 3 mutaciones redundantes finalmente se obtuvieron 555 mutantes simples obtenidas desde

AB-BIND. Desde SKEMPI2.0 se obtuvieron un total de 696 mutantes simples que comprenden 50 complejos Ag-Ab diferentes.

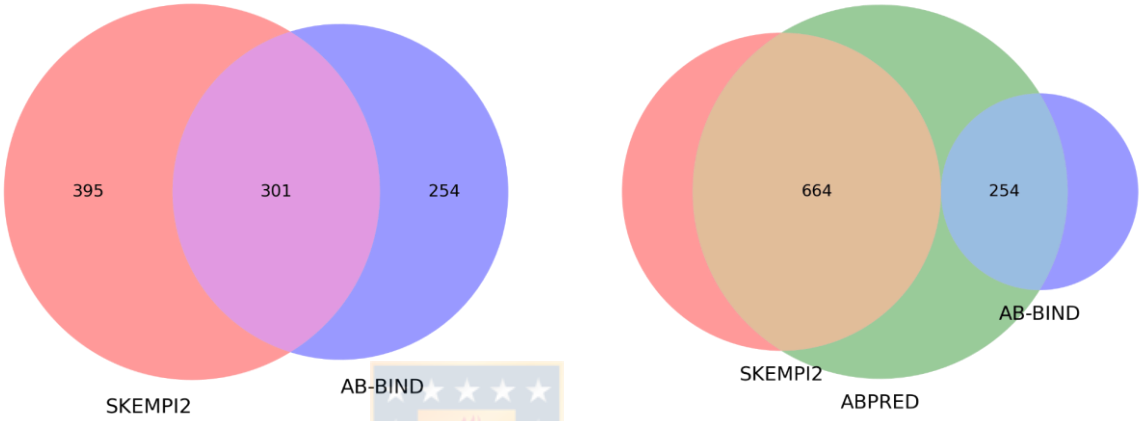

Figura 9. Diagrama de Venn del número de mutantes en AB-BIND, SKEMPI2.0 y ABPRED_DB.

En la Figura 9 se observa los resultados del análisis de conjuntos entre AB-BIND y SKEMPI2.0. Se encontraron 301 mutantes compartidas (intersección entre ambos conjuntos), la unión de ambos conjuntos da un total de 950 mutantes. Al unir ambos conjuntos se eliminaron posibles mutantes duplicadas quedando un total de 918 mutantes en 57 complejos antígeno-anticuerpo, formando la base de datos ABPRED_DB.

(https://github.com/victorfica/Master-thesis/blob/master/data/ABPRED_DB.csv).

4.2 Conjunto de datos ABPRED_DATA

A partir de las propiedades calculadas mediante Arpeggio, POPS y FoldX, se generó un total de 77 descriptores que engloban características de las interacciones no covalentes de la interfaz, el SASA de las estructuras mutantes y su interfaz, y términos energéticos (Tabla 2). Además, se modelaron las hipotéticas mutantes inversas, esto resultó en un conjunto inicial ABPRED_DATA, una matriz de datos de 1836 x 77 (ejemplo en Tabla 3). En la Figura 10 podemos ver la distribución balanceada de los valores de ΔΔG al tener las retromutantes modeladas en el conjunto de datos.

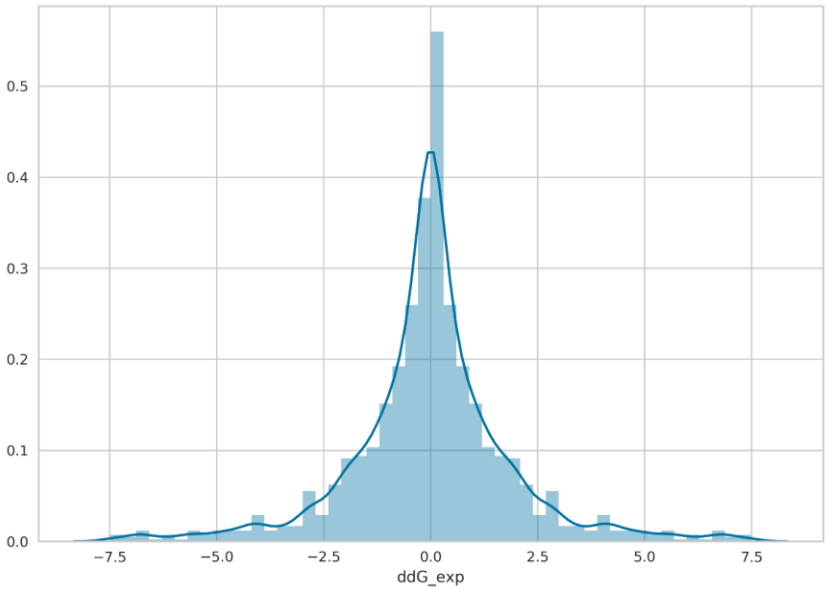

Figura 10. Distribución de valores ΔΔG en ABPRED_DATA.

Tabla 2. Características utilizadas para modelar ABPRED.

Tipo de característica	**Características**	**ID característica/ Total**	**Herramienta**
Interacciones no covalentes de la interfaz	Número de contactos en la interfaz mutante, diferencia del número de contactos entre interfaz mutante y retromutante.	clash, covalent, vdw_clash, vdw, proximal, hbond, weak_hbond, xbond, ionic, metal_complex, aromatic, hydrophobic, carbonyl, polar, weak_polar. **Total: 15**	Arpeggio, ABPRED
SASA de interfaz	SASA total, hidrofílico e hidrofóbico del complejo mutante y el SASA total, hidrofílico e hidrofóbico de la interfaz mutante.	if_phil, if_phob, if_total, sasa_phil, sasa_phob, sasa_total. **Total: 6**	POPS, ABPRED
Términos energéticos	Diferencia de los términos energéticos entre estructura mutante y retromutante, Términos energéticos de la estructura mutante, ΔΔG.	dg_change, intraclashes_energy_1_change, intraclashes_energy_2_change, backbone_hbond_change, sidechain_hbond_change, van_der_waals_change, electrostatics_change, solvation_polar_change, solvation_hydrophobic_change, van_der_waals_clashes_change, entropy_sidechain_change, entropy_mainchain_change, sloop_entropy_change, mloop_entropy_change, cis_bond_change, torsional_clash_change, backbone_clash_change, helix_dipole_change, water_bridge_change, disulfide_change, electrostatic_kon_change, partial_covalent_bonds_change, energy_ionisation_change, entropy_complex_change, interface_residues_change, interface_residues_clashing_change, interface_residues_vdw_clashing_change, interface_residues_bb_clashing_change, intraclashes_energy_1_mut, intraclashes_energy_2_mut, dg_mut, backbone_hbond_mut, sidechain_hbond_mut,	Foldx, ABPRED

		van_der_waals_mut, electrostatics_mut, solvation_polar_mut, solvation_hydrophobic_mut, van_der_waals_clashes_mut, entropy_sidechain_mut, entropy_mainchain_mut, sloop_entropy_mut, mloop_entropy_mut, cis_bond_mut, torsional_clash_mut, backbone_clash_mut, helix_dipole_mut, water_bridge_mut, disulfide_mut, electrostatic_kon_mut, partial_covalent_bonds_mut, energy_ionisation_mut, entropy_complex_mut, interface_residues_mut, interface_residues_clashing_mut, interface_residues_vdw_clashing_mut, interface_residues_bb_clashing_mut. **Total: 56**	

Tabla 3. Muestra parcial de datos ABPRED_DATA.

index	clash	covalent	vdw_clash	vdw	proximal	hbond	weak_hbond	xbond	ionic	metal_complex	...
1AK4.DA488G.Repair2.clean.mut.pdb	0.0	0.0	7.0	6.0	282.0	7.0	2.0	0.0	0.0	0.0	...
1AK4.DA488V.Repair2.clean.mut.pdb	0.0	0.0	13.0	7.0	335.0	7.0	5.0	0.0	0.0	0.0	...
1AK4.DA492G.Repair2.clean.mut.pdb	0.0	0.0	9.0	5.0	302.0	7.0	3.0	0.0	0.0	0.0	...
1AK4.DA492V.Repair2.clean.mut.pdb	0.0	0.0	8.0	7.0	321.0	7.0	3.0	0.0	0.0	0.0	...
1AK4.DG489A.Repair2.clean.mut.pdb	0.0	0.0	9.0	6.0	307.0	7.0	3.0	0.0	0.0	0.0	...
1AK4.DG489V.Repair2.clean.mut.pdb	0.0	0.0	9.0	5.0	314.0	7.0	3.0	0.0	0.0	0.0	...
1AK4.DH487A.Repair2.clean.mut.pdb	0.0	0.0	7.0	3.0	278.0	6.0	3.0	0.0	0.0	0.0	...
1AK4.DH487Q.Repair2.clean.mut.pdb	0.0	0.0	8.0	3.0	291.0	7.0	2.0	0.0	0.0	0.0	...
1AK4.DH487R.Repair2.clean.mut.pdb	0.0	0.0	8.0	4.0	305.0	7.0	4.0	0.0	0.0	0.0	...
1AK4.DI491A.Repair2.clean.mut.pdb	0.0	0.0	8.0	6.0	308.0	7.0	3.0	0.0	0.0	0.0	...

4.3 Modelamiento de ABPRED

En esta etapa se utilizaron los datos ABPRED_DATA junto a los datos experimentales de ΔΔG, para entrenar y evaluar un predictor de ΔΔG basado en la metodología de *stacking*. Como parte del proceso primero se separaron los datos de entrenamiento y prueba, ABPRED_DATA80 y ABPRED_DATA20 respectivamente. La Figura 11 muestra el resultado de la separación utilizando muestreo estratificado aleatorio, al aplicar binarización en la variable objetivo ΔΔG.

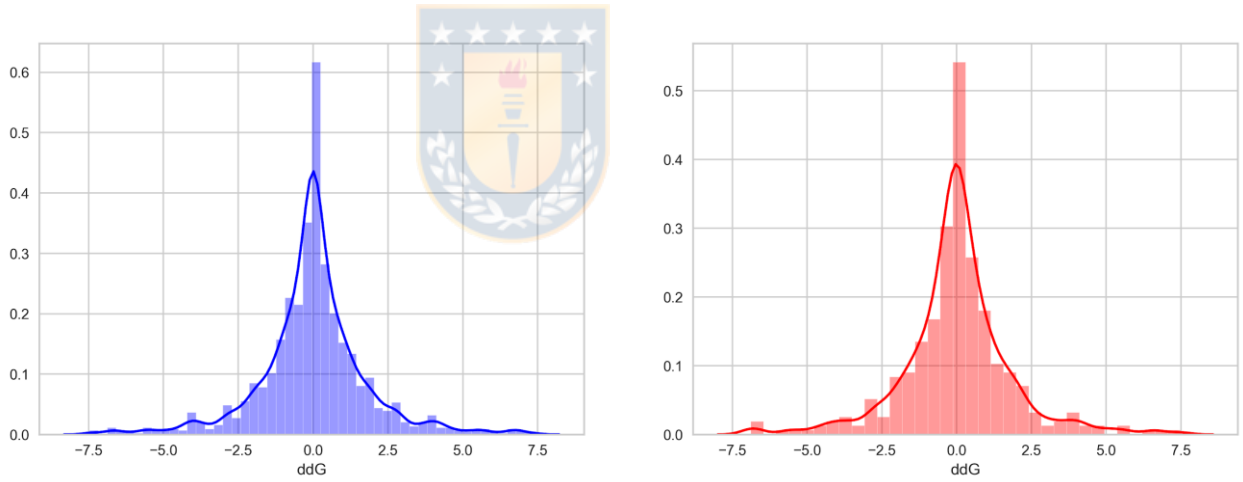

Figura 11. Distribución de variable ΔΔG en datos de entrenamiento (azul) y prueba (rojo).

Con el objetivo de tener una evaluación robusta se realizaron las 2 metodologías de evaluación generalmente recomendadas: Una validación cruzada con ABPRED_DATA80 y una evaluación independiente con ABPRED_DATA20. Para

la validación cruzada de 10 iteraciones, nuestro modelo alcanzó una **Correlación de Pearson** (ρ) de 0.6 con una desviación estándar (σ) de 0.0456, un **RMSE** de 1.4568 Kcal/mol (σ = 0.1469) y un **R^2** de 0.3503 (σ = 0.0546).

La Figura 12 muestra los resultados de ABPRED en la evaluación independiente con ABPRED_DATA20, alcanzando una Correlación de Pearson de 0.55, un **RMSE** de 1.49 Kcal/mol y un **R^2** de 0.302.

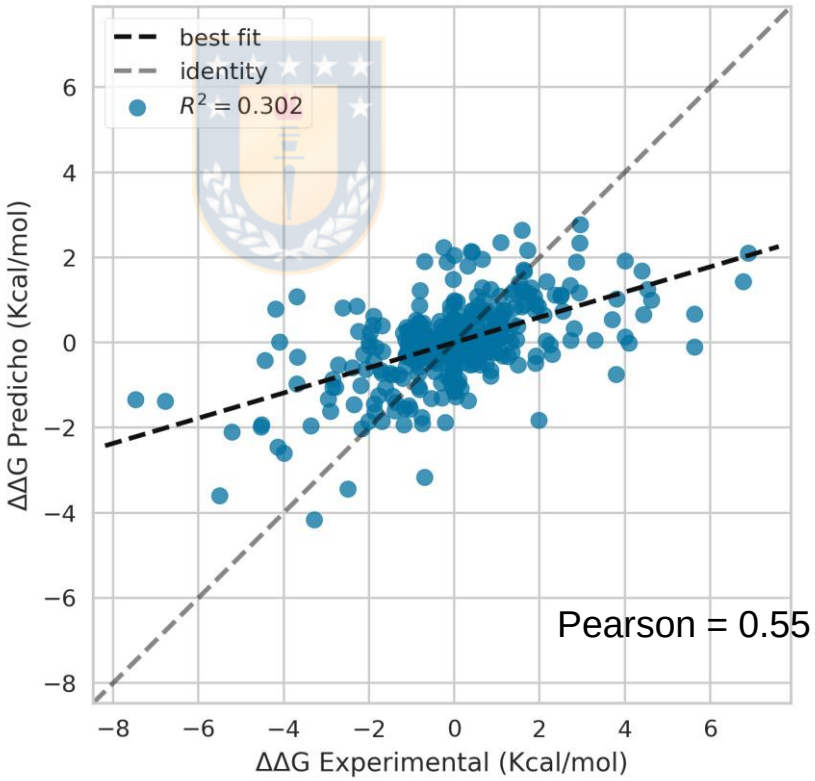

Figura 12. Rendimiento de ABPRED en predecir cambios de afinidad en Ag-Ab dado mutaciones puntuales.
ABPRED alcanza una Correlación de Pearson de 0.55, un RMSE de 1.49 Kcal/mol y un R^2 de 0.302, en la evaluación independiente usando ABPRED_DATA20.

4.4 Rendimiento comparativo de *ABPRED*

Este último análisis tuvo como objetivo comparar el rendimiento predictivo de ABPRED con otros predictores representativos de diferentes bases teóricas. Para ello se utilizó el conjunto ciego de datos, ADAPT36, obtenidos desde el trabajo publicado por Vivcharuk et al. 2017.

Este conjunto de datos presentó una particularidad por lo cual el rendimiento de ABPRED fue irregular en comparación a los demás métodos. Una de las mutantes, específicamente 3BE1.HY33R, posee un ΔΔG experimental reportado de 3.06, lo que escapa por mucho del promedio (-0.263243). La correlación de Pearson observada en ABPRED fue negativa en comparación a los demás predictores, y se debe principalmente a esta mutante (ver Figura 13A). Los valores de correlación y RMSE están reportados en la Tabla 4, y vemos que, a pesar de obtener correlación negativa, el RMSE de ABPRED es similar al del resto de predictores (1.0 Kcal/mol). Al eliminar la mutante 3BE1.HY33R de ADAPT36, el rendimiento de ABPRED mejora notablemente, de -0.316 a 0.257 en correlación de Pearson y de un RMSE de 1.0 a 0.552 Kcal/mol (Tabla 4). De esta forma ABPRED logra superar en RMSE al resto de los predictores y su correlación es similar al resto. En la Figura 13B, en línea amarilla, se observa como el ajuste lineal cambia totalmente al eliminar 3BE1.HY33R.

Tabla 4. Rendimiento comparativo de predictores representativos disponibles y ABPRED.

Predictores	Correlación de Pearson	RMSE (Kcal/mol)
DFIRE	0.336/0.29[a]	0.943/0.853[a]
FoldX	0.268/-0.221[a]	0.937/0.94[a]
BeatMusic	0.584/0.276[a]	0.874/0.843[a]
mCSB-AB	0.45/0.242[a]	0.878/0.868[a]
ABPRED	**-0.316/0.257[a]**	**1.0/0.552**[a]

[a] Rendimiento descartando la mutante 3BE1.HY33R

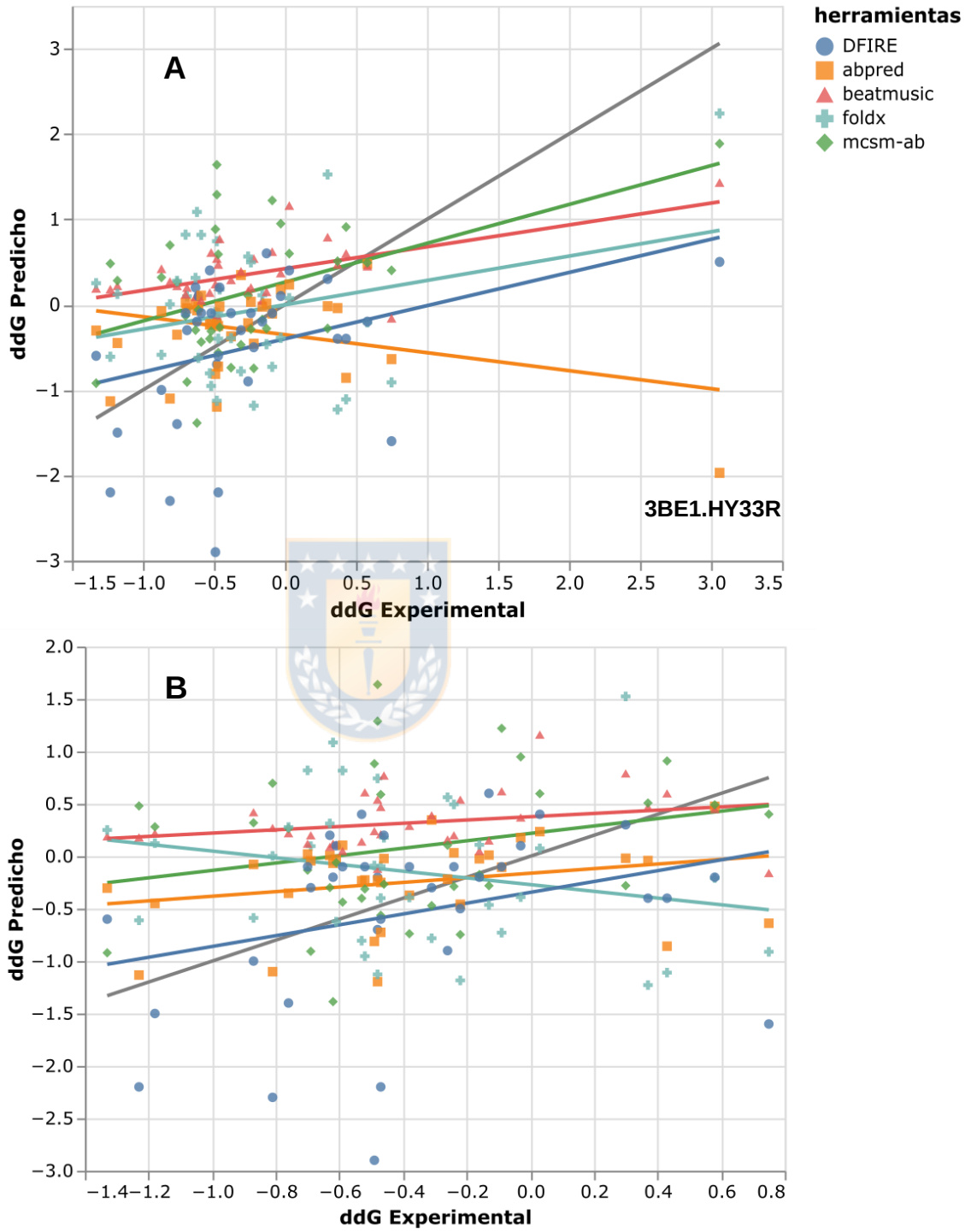

Figura 13. Gráfica de dispersión para las predicciones de cada método.
En gris el ajuste lineal que corresponde a una predicción perfecta y en colores el ajuste lineal para cada método. A) resultados de predictores en datos ADAPT36, B) resultados de predictores al eliminar mutante 3BE1.HY33R

5 DISCUSIÓN

El aumento en la cantidad de información estructural sobre anticuerpos, la creación de nuevas bases de datos especializadas y la aparición de herramientas bioinformáticas que utilizan toda esta información, han sido el resultado del gran interés que existe por el diseño de anticuerpos, en especial por su potencial terapéutico.

Bases de datos como PROXiMATE (Jemimah, Yugandhar, and Michael Gromiha 2017), SKEMPI2 y AB-BIND han permitido explorar el desarrollo de nuevos predictores basados en aprendizaje automático enfocados en la tarea de predecir la afinidad de unión o los cambios en la afinidad de unión, ΔΔG. La ventaja de estos métodos es que permiten una predicción de los cambios de afinidad de alto rendimiento, a diferencia de métodos más rigurosos como FEP, los cuales son computacionalmente más costosos. Esto los vuelve atractivos para abordar problemas como el de la maduración de la afinidad *in silico* de anticuerpos, que requerirían explorar un gran número de combinaciones en sustituciones aminoacídicas.

En este contexto ABPRED se posicionaría como una alternativa que ha demostrado tener rendimiento similar o superior a métodos clásicos como FoldX

y DFIRE, pero también compitiendo con métodos recientes como mCSM-AB uno de los pocos predictores de ΔΔG específico para complejos antígeno-anticuerpo. La base de datos ABPRED_DB construida al unir los datos obtenidos desde SKEMPI2.0 y AB-BIND, al considerar posibles mutantes redundantes debido a diferencias en la ID de cadenas de un mismo complejo, permitió expandir el número actualmente conocido de mutaciones simples en complejos antígeno-anticuerpo a 918. Esto permitió tener un conjunto de 1836 mutantes y retromutantes, de los cuales 1425 fueron usados para entrenar ABPRED.

5.1 ABPRED y características

Conocer las características más importantes utilizadas para predecir el impacto de una mutación sobre la afinidad de unión de un complejo, debería contribuir a nuestro entendimiento de los mecanismos subyacentes en las interacciones proteína-proteína. En los predictores lineales clásicos de ΔΔG, cada peso en los componentes de la función lineal usualmente refleja su importancia relativa. En contraste, para modelos de aprendizaje automático es más complejo extraer la importancia de cada característica debido a la naturaleza no-lineal de los algoritmos. Existen algoritmos que pueden ranquear las características basados en su contribución relativa al modelo final, tales como árboles de decisiones o Random Forest. ELASPIC (Berliner et al. 2014) es un predictor ΔΔG que utiliza el algoritmo Gradient Boosted Trees, el cual le permite reportar la importancia

relativa de las características utilizadas. Dentro de ellas, el ΔΔG de FoldX fue una de las más importantes, seguido por otros términos energéticos tales como *sidechain entropy, energy clashes, solvation energy.* De manera similar BindProfX (Xiong et al. 2017), también reportó como principal característica el ΔΔG de FoldX seguido de los perfiles de interfaz.

En la evaluación ciega ABPRED se comparó junto a mCSM-AB, otro método basado en aprendizaje automático específico para complejos antígeno-anticuerpo. A diferencia de ABPRED, mCSM-AB no incluye características energéticas. Al comparar los resultados de ABPRED versus mCSM-AB en la evaluación con ADAPT36, se observa un menor error (RMSE de 0.552 y 0.868 Kcal/mol respectivamente). Esto podría ser evidencia de la importancia de incluir características energéticas, ya que los valores de correlación fueron muy similares pero el error fue menor en ABPRED.

Estos análisis muestran la importancia de las energías intermoleculares para estimar los efectos de mutaciones en la afinidad de unión, en combinación con otras características. Esto también explicaría el sobresaliente rendimiento de ABPRED por sobre otros métodos clásicos al incluir características energéticas. Si bien con la metodología *stacking* aplicada en ABPRED, no es posible ranquear la importancia de las características en el modelo final, un análisis previo de correlación entre características y el ΔΔG experimental muestra que dentro de las 10 características con más alta correlación, la mayoría son componentes energéticos obtenidos desde FoldX (ver Figura 15).

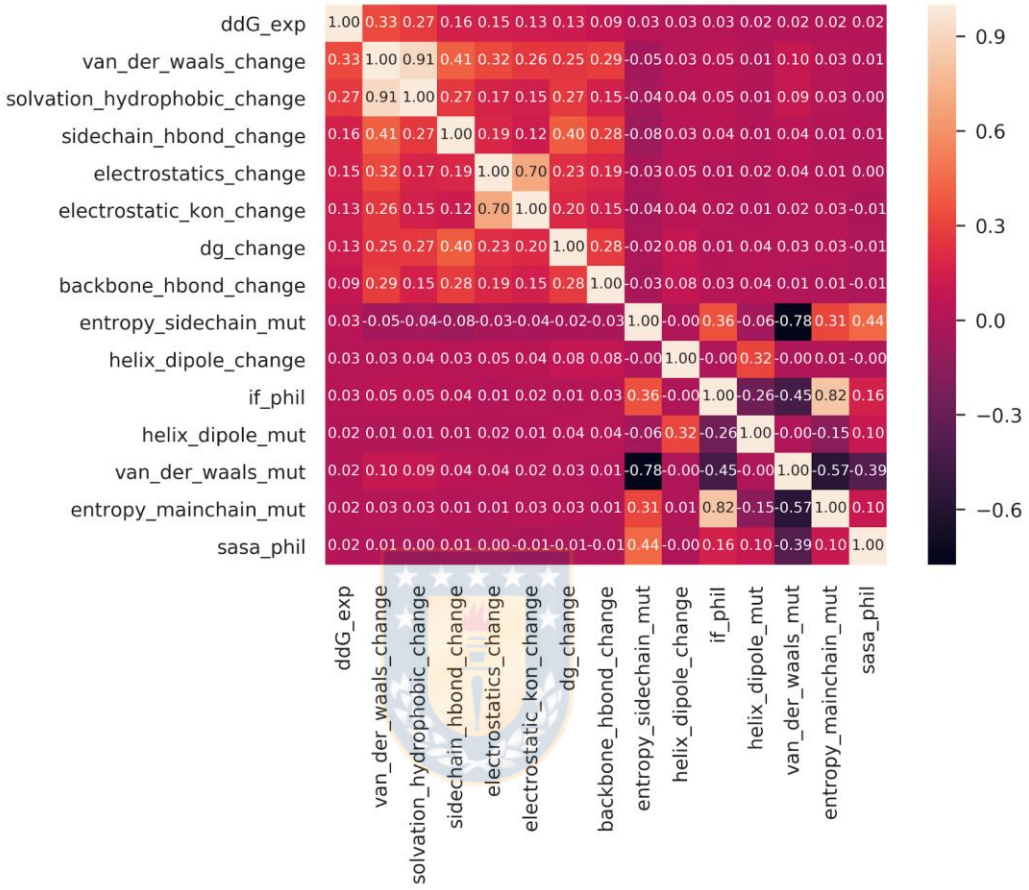

Figura 14. Mapa de calor de correlación de pares entre las primeras 14 características.
Valores representan la correlación de Spearman. Se incluye la variable experimental ΔΔG en la primera fila**.**

5.2 Evaluación ciega

Los resultados de la evaluación comparativa pusieron en evidencia que aún queda mucho espacio para mejorar en los predictores de ΔΔG. Ya se ha visto en general que cuando los predictores son evaluados en datos experimentales

nuevos, que no hayan visto al entrenar sus modelos, estos tienen dificultades. Por ejemplo, cuando se publicó la versión 2 de SKEMPI, en un análisis comparativo con un subconjunto ciego (487 mutaciones en 56 complejos), el cual ninguno de los predictores vio previamente, mostró que iSEE (Geng, Vangone, et al. 2019), FoldX, mCSM (Pires et al., 2014) y BindProfX, no tuvieron tan buenos resultados como en sus conjuntos de entrenamiento, con valores de Correlación de Pearson todos bajo 0.4.

En nuestra evaluación ciega con ADAPT36, si bien el conjunto de datos es pequeño, resalta la baja correlación de todos los predictores. En la Tabla 4 se puede observar que todos los predictores, excepto BeatMusic, obtuvieron una correlación igual o menor a 0.45 y un RMSE entre 0.87 – 0.94 Kcal/mol en ADAPT36. La mutante 3BE1.HY33R fue particularmente problemática para ABPRED, cuyo valor de ΔΔG predicho fue tan malo en relación ΔΔG experimental, que causo que la correlación obtenida fuese negativa (-0.316). En la Figura 13 se observa el efecto de eliminar 3BE1.HY33R del conjunto ADAPT36 (ADAPT36 filtrado) y el efecto en las líneas de ajuste graficadas (Figura 13B). El impacto se ve reflejado además en el aumento del valor de correlación de Pearson a 0.257.

La causa en la dificultad de ABPRED para predecir esta mutante pudiese estar en baja cantidad de mutaciones en ABPRED_DATA con valores ΔΔG experimentales cercanos a 3.0 Kcal/mol. Los pocos datos cercanos a esos valores, son un componente directo de sesgo al momento de entrenar ABPRED.

El aumento de información experimental para entrenar modelos basados en aprendizaje automático es crucial en este sentido.

Finalmente, como se observa en la Tabla 3, los resultados obtenidos con ADAPT36 filtrado, destacan a ABPRED en su bajo RMSE (0.552 Kcal/mol) superando a los demás métodos, y una correlación similar al resto de los métodos.

5.3 Clasificando la dirección del cambio de afinidad

Otra forma de abordar el problema de predicción de ΔΔG, es verlo como un problema de clasificación binaria. Muchas funciones de puntaje enfocadas docking molecular tienen este enfoque conceptual, ya que parte central del docking se basa en identificar ligandos de no-ligandos (Cheng et al., 2009).

En este contexto quisimos explorar, de forma simple, la capacidad de ABPRED más allá de la predicción de afinidades medidas experimentalmente, en particular si ABPRED es capaz de identificar cuando una mutante es de escape o no. Para esto utilizamos el anticuerpo terapéutico anti-VIH VRC01, el cual reconoce la glicoproteína gp120 de HIV-1. Anteriormente ya se ha estudiado experimentalmente el efecto de 78 mutaciones en gp120 en la efectividad de VRC01 (Li et al., 2011, p. 1). Dentro de estas mutaciones evaluadas, 8 disminuyen la afinidad y 2 mutaciones la aumentan. ABPRED logro clasificar la dirección del cambio de afinidad correctamente en 9 de 10 mutaciones (aumento

de afinidad: 8/8, disminución afinidad: 1/2). Estos resultados sugieren el potencial poder predictivo de este enfoque para explorar las consecuencias de mutaciones clínica o biológicamente relevantes. Como dato relevante, el tiempo que demora ABPRED en evaluar las 78 mutaciones es de ~65 minutos (Laptop con CPU AMD Ryzen 5 4600H).

5.4 Limitaciones y Proyecciones

La disponibilidad de estructuras experimentales de complejos proteicos es limitada, y más aún para el subgrupo antígeno-anticuerpo, es por esto que se recurre a métodos de modelamiento comparativo. Pero para predecir el impacto de las mutaciones a una escala de interactoma, se hace necesario desarrollar y/o evaluar predictores de ΔΔG capaces de manejar estructuras modeladas. No todos los predictores son entrenados tomando esto en consideración, y esta también es una de las limitaciones de ABPRED. Nuestro método no fue entrenado ni evaluado contra modelos comparativos. Además, los intentos por utilizar modelos estructurales para predecir ΔΔG han demostrado que la calidad de los modelos es crucial para alcanzar buen rendimiento predictivo (Xiong et al. 2017; Dourado and Flores 2016).

Una proyección inmediata de este trabajo es la integración de ABPRED en un flujo de mutagénesis utilizando diferentes esquemas de sustitución (trabajo en

progreso). La Figura 16 ejemplifica las capacidades que tiene ABPRED de realizar esquemas de mutagénesis *in silico* a lo largo de cada CDR y reportar aquellas sustituciones que aumentan la afinidad en cada posición (barras rojas). Existen diferentes esquemas de mutagénesis propuestos en los cuales ABPRED puede ser integrado, entre estos tenemos: a) Reductivo (Ser, Tyr, Gly, Phe. Asp, Asn); b) Extendido (Asp, Glu, Phe, Gly, Ile, Leu, Lys, Asn, Gln, Arg, Ser, Thr, Val, Tyr); c) KMT (Ala, Asp, Ser o Tyr); d) WMC (Asn, Ser, Thr, o Tyr); y e) RRT (Asn, Asp, Gly, or Ser). Los primeros 2 están basados en el análisis de diversidad de aminoácidos en los CDR (Clark et al., 2006), y los últimos 3 basados en el análisis de diversidad de los CDR por metodologías de librerías sintéticas (Shim, 2015). Otra proyección, la cual puede ser una extensión a lo propuesto anteriormente, es integrar cálculos rigurosos de dinámica molecular y metodologías como FEP (Free energy Perturbation). Esto involucraría 3 pasos: 1) realizar mutagénesis con ABPRED usando alguno de los esquemas propuestos; 2) Agrupar y ranquear las mutaciones con efectos estabilizantes y 3) Calcular el cambio de energía libre en un número reducido de las mutantes seleccionadas usando FEP.

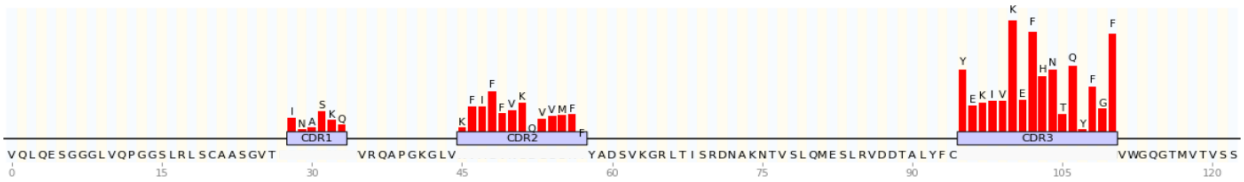

Figura 15. Ejemplo de resultados obtenidos por mutagénesis saturante utilizando ABPRED. Datos no publicados.

6 CONCLUSIONES

Se logró obtener la base de datos ABPRED_DB que consta de 918 mutantes simples en 57 complejos antígeno-anticuerpo, además de sus respectivas constantes cinéticas experimentalmente calculadas.

Se implementaron herramientas para el cálculo de características que modelan aspectos estructurales y energéticos de la interacción antígeno-anticuerpo y sus efectos en mutaciones. Con esto se logró generar un conjunto de datos ABPRED_DATA con una dimensión de 1837 x 77.

Se entrenó y evaluó un modelo de aprendizaje automático llamado ABPRED utilizando ABPRED_DATA80 (datos de entrenamiento) y ABPRED_DATA20 (datos de prueba) respectivamente. ABPRED alcanzó una correlación de Pearson de 0.6 (RMSE = 1.457) y 0.55 (RMSE = 1.49), en el conjunto de entrenamiento y prueba respectivamente, demostrando un mejor rendimiento a métodos clásicos que han sido previamente usados para diseñar anticuerpos

La evaluación comparativa (evaluación ciega) demostró mejor precisión que los demás predictores con un RMSE de 0.552 Kcal/mol (ADAPT36 filtrado), y una correlación similar con un valor de correlación de Pearson de 0.257. Al mismo tiempo se evidenció el bajo rendimiento general que los predictores de

ΔΔG específicos de antígeno-anticuerpo mostraron con los datos ADAPT36 filtrados.

Por último, mencionar que todo este trabajo involucró el desarrollo de una herramienta en Python llamada Abpred, la cual está disponible en https://github.com/DatagenCL/AbPred. Esta herramienta es un programa con una interfaz de línea de comandos que permite correr predicciones de ΔΔG dada una lista de mutaciones y una estructura PDB de un complejo Ag-Ab.

8 BIBLIOGRAFÍA

Abbas, Abul K., and Andrew H. Lichtman. Basic Immunology: Functions and Disorders of the Immune System. Philadelphia, PA: Elsevier Saunders, 2006.

Almagro, J. C., & Fransson, J. (2008). Humanization of antibodies. Frontiers in Bioscience: A Journal and Virtual Library, 13, 1619–1633.

Ambrosetti, F., Jiménez-García, B., Roel-Touris, J., & Bonvin, A. M. J. J. (2019). Modeling Antibody-Antigen Complexes by Information-Driven Docking. Structure, 0(0). https://doi.org/10.1016/j.str.2019.10.011

Antibody Society 2019. Approved antibodies. Available at https://www.antibodysociety.org/resources/approved-antibodies/. Consultado en Junio 17, 2019.

Barderas, R., Desmet, J., Timmerman, P., Meloen, R., & Casal, J. I. (2008). Affinity maturation of antibodies assisted by in silico modeling. Proceedings of the National Academy of Sciences of the United States of America, 105(26), 9029–9034. https://doi.org/10.1073/pnas.0801221105

Berliner, N., Teyra, J., Çolak, R., Garcia Lopez, S., & Kim, P. M. (2014). Combining Structural Modeling with Ensemble Machine Learning to Accurately Predict Protein Fold Stability and Binding Affinity Effects upon Mutation. PLoS ONE, 9(9). https://doi.org/10.1371/journal.pone.0107353

Bradbury, A. R. M., Sidhu, S., Dübel, S., & McCafferty, J. (2011). Beyond natural antibodies: The power of in vitro display technologies. Nature Biotechnology, 29(3), 245–254. https://doi.org/10.1038/nbt.1791

Cavallo, L., Kleinjung, J., & Fraternali, F. (2003). POPS: A fast algorithm for solvent accessible surface areas at atomic and residue level. Nucleic Acids Research, 31(13), 3364–3366. https://doi.org/10.1093/nar/gkg601

Chan, K. T. (1999). Humanization, expression and characterization of an anti-hepatocellular carcinoma monoclonal antibody.

Cheng, T., Li, X., Li, Y., Liu, Z., & Wang, R. (2009). Comparative Assessment of Scoring Functions on a Diverse Test Set. Journal of Chemical Information and Modeling, 49(4), 1079–1093. https://doi.org/10.1021/ci9000053

Clark, L. A., Boriack-Sjodin, P. A., Eldredge, J., Fitch, C., Friedman, B., Hanf, K. J. M., Jarpe, M., Liparoto, S. F., Li, Y., Lugovskoy, A., Miller, S., Rushe, M., Sherman, W., Simon, K., & Van Vlijmen, H. (2006). Affinity enhancement of an in vivo matured therapeutic antibody using structure-based computational design. Protein Science : A Publication of the Protein Society, 15(5), 949–960. https://doi.org/10.1110/ps.052030506

Clark, L. A., Ganesan, S., Papp, S., & Vlijmen, H. W. T. van. (2006). Trends in Antibody Sequence Changes during the Somatic Hypermutation Process. The Journal of Immunology, 177(1), 333–340. https://doi.org/10.4049/jimmunol.177.1.333

Dahiyat, B. I., & Mayo, S. L. (1997). Probing the role of packing specificity in protein design. Proceedings of the National Academy of Sciences, 94(19), 10172–10177. https://doi.org/10.1073/pnas.94.19.10172

Dehouck, Y., Kwasigroch, J. M., Rooman, M., & Gilis, D. (2013). BeAtMuSiC: Prediction of changes in protein-protein binding affinity on mutations. Nucleic Acids Research, 41(Web Server issue), W333-339. https://doi.org/10.1093/nar/gkt450

Dourado, D. F. A. R., & Flores, S. C. (2014). A multiscale approach to predicting affinity changes in protein-protein interfaces. Proteins, 82(10), 2681–2690. https://doi.org/10.1002/prot.24634

Dourado, D. F. A. R., & Flores, S. C. (2016). Modeling and fitting protein-protein complexes to predict change of binding energy. Scientific Reports, 6(1), 25406. https://doi.org/10.1038/srep25406

Dunbrack, R. L. (2002). Rotamer libraries in the 21st century. Current Opinion in Structural Biology, 12(4), 431–440.

Friedman, J. H. (2002). Stochastic gradient boosting. Computational Statistics & Data Analysis, 38(4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2

Friedman, J. H., Hastie, T., & Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software, 33(1), 1–22. https://doi.org/10.18637/jss.v033.i01

Geng, C., Vangone, A., Folkers, G. E., Xue, L. C., & Bonvin, A. M. J. J. (2019). iSEE: Interface structure, evolution, and energy-based machine learning predictor of binding affinity changes upon mutations. Proteins: Structure, Function, and Bioinformatics, 87(2), 110–119. https://doi.org/10.1002/prot.25630

Geng, C., Xue, L. C., Roel-Touris, J., & Bonvin, A. M. J. J. (2019). Finding the

ΔΔG spot: Are predictors of binding affinity changes upon mutations in protein–protein interactions ready for it? Wiley Interdisciplinary Reviews: Computational Molecular Science, 0(0), e1410. https://doi.org/10.1002/wcms.1410

Harder, T., Boomsma, W., Paluszewski, M., Frellsen, J., Johansson, K. E., & Hamelryck, T. (2010). Beyond rotamers: A generative, probabilistic model of side chains in proteins. BMC Bioinformatics, 11(1), 306. https://doi.org/10.1186/1471-2105-11-306

Hu, D., Hu, S., Wan, W., Xu, M., Du, R., Zhao, W., Gao, X., Liu, J., Liu, H., & Hong, J. (2015). Effective Optimization of Antibody Affinity by Phage Display Integrated with High-Throughput DNA Synthesis and Sequencing Technologies. PLOS ONE, 10(6), e0129125. https://doi.org/10.1371/journal.pone.0129125

Jakobovits, A. (1995). Production of fully human antibodies by transgenic mice. Current Opinion in Biotechnology, 6(5), 561–566.

Jankauskaite, J., Jiménez-García, B., Dapkunas, J., Fernández-Recio, J., & Moal, I. H. (2019). SKEMPI 2.0: An updated benchmark of changes in protein-protein binding energy, kinetics and thermodynamics upon mutation. Bioinformatics (Oxford, England), 35(3), 462–469. https://doi.org/10.1093/bioinformatics/bty635

Jemimah, S., Yugandhar, K., & Michael Gromiha, M. (2017). PROXiMATE: A database of mutant protein-protein complex thermodynamics and kinetics. Bioinformatics (Oxford, England), 33(17), 2787–2788. https://doi.org/10.1093/bioinformatics/btx312

Jubb, H. C., Higueruelo, A. P., Ochoa-Montaño, B., Pitt, W. R., Ascher, D. B., & Blundell, T. L. (2017). Arpeggio: A Web Server for Calculating and Visualising Interatomic Interactions in Protein Structures. Journal of Molecular Biology, 429(3), 365–371. https://doi.org/10.1016/j.jmb.2016.12.004

Kiyoshi, M., Caaveiro, J. M. M., Miura, E., Nagatoishi, S., Nakakido, M., Soga, S., Shirai, H., Kawabata, S., & Tsumoto, K. (2014). Affinity Improvement of a Therapeutic Antibody by Structure-Based Computational Design: Generation of Electrostatic Interactions in the Transition State Stabilizes the Antibody-Antigen Complex. PLoS ONE, 9(1), e87099. https://doi.org/10.1371/journal.pone.0087099

Koga, N., Tatsumi-Koga, R., Liu, G., Xiao, R., Acton, T. B., Montelione, G. T., & Baker, D. (2012). Principles for designing ideal protein structures. Nature, 491(7423), 222–227. https://doi.org/10.1038/nature11600

Köhler, G., & Milstein, C. (1975). Continuous cultures of fused cells secreting antibody of predefined specificity. Nature, 256(5517), 495.

https://doi.org/10.1038/256495a0

Kuhlman, B., Dantas, G., Ireton, G. C., Varani, G., Stoddard, B. L., & Baker, D. (2003). Design of a Novel Globular Protein Fold with Atomic-Level Accuracy. Science, 302(5649), 1364–1368. https://doi.org/10.1126/science.1089427

Li, Y., O'Dell, S., Walker, L. M., Wu, X., Guenaga, J., Feng, Y., Schmidt, S. D., McKee, K., Louder, M. K., Ledgerwood, J. E., Graham, B. S., Haynes, B. F., Burton, D. R., Wyatt, R. T., & Mascola, J. R. (2011). Mechanism of neutralization by the broadly neutralizing HIV-1 monoclonal antibody VRC01. Journal of Virology, 85(17), 8954–8967. https://doi.org/10.1128/JVI.00754-11

Lippow, S. M., Wittrup, K. D., & Tidor, B. (2007). Computational design of antibody-affinity improvement beyond in vivo maturation. Nature Biotechnology, 25(10), 1171–1176. https://doi.org/10.1038/nbt1336

Moal, I. H., & Fernández-Recio, J. (2012). SKEMPI: A Structural Kinetic and Energetic database of Mutant Protein Interactions and its use in empirical models. Bioinformatics (Oxford, England), 28(20), 2600–2607. https://doi.org/10.1093/bioinformatics/bts489

Moal, I. H., & Fernandez-Recio, J. (2013). Intermolecular Contact Potentials for Protein-Protein Interactions Extracted from Binding Free Energy Changes upon Mutation. Journal of Chemical Theory and Computation, 9(8), 3715–3727. https://doi.org/10.1021/ct400295z

Moal, I. H., Moretti, R., Baker, D., & Fernández-Recio, J. (2013). Scoring functions for protein-protein interactions. Current Opinion in Structural Biology, 23(6), 862–867. https://doi.org/10.1016/j.sbi.2013.06.017

Mosca, R., Céol, A., & Aloy, P. (2013). Interactome3D: Adding structural details to protein networks. Nature Methods, 10(1), 47–53. https://doi.org/10.1038/nmeth.2289

Nagi, S., & Bhattacharyya, D. Kr. (2013). Classification of microarray cancer data using ensemble approach. Network Modeling Analysis in Health Informatics and Bioinformatics, 2(3), 159–173. https://doi.org/10.1007/s13721-013-0034-x

North, B., Lehmann, A., & Dunbrack Jr, R. L. (2011). A New Clustering of Antibody CDR Loop Conformations. Journal of Molecular Biology, 406(2), 228–256. https://doi.org/10.1016/j.jmb.2010.10.030

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn:

Machine Learning in Python. Journal of Machine Learning Research, 12, 2825−2830.

Pires, D. E. V., & Ascher, D. B. (2016). mCSM-AB: A web server for predicting antibody–antigen affinity changes upon mutation with graph-based signatures. Nucleic Acids Research, 44(W1), W469–W473. https://doi.org/10.1093/nar/gkw458

Pires, D. E. V., Ascher, D. B., & Blundell, T. L. (2014). mCSM: Predicting the effects of mutations in proteins using graph-based signatures. Bioinformatics (Oxford, England), 30(3), 335–342. https://doi.org/10.1093/bioinformatics/btt691

Rodrigues, C. H. M., Myung, Y., Pires, D. E. V., & Ascher, D. B. (2019). mCSM-PPI2: Predicting the effects of mutations on protein-protein interactions. Nucleic Acids Research, 47(W1), W338–W344. https://doi.org/10.1093/nar/gkz383

Sela-Culang, I., Kunik, V., & Ofran, Y. (2013). The structural basis of antibody-antigen recognition. Frontiers in Immunology, 4, 302. https://doi.org/10.3389/fimmu.2013.00302

Selzer, T., Albeck, S., & Schreiber, G. (2000). Rational design of faster associating and tighter binding protein complexes. Nature Structural Biology, 7(7), 537. https://doi.org/10.1038/76744

Shanthirabalan, S., Chomilier, J., & Carpentier, M. (2018). Structural effects of point mutations in proteins. Proteins: Structure, Function, and Bioinformatics, 86(8), 853–867. https://doi.org/10.1002/prot.25499

Shim, H. (2015). Synthetic approach to the generation of antibody diversity. BMB Reports, 48(9), 489–494. https://doi.org/10.5483/bmbrep.2015.48.9.120

Shrake, A., & Rupley, J. A. (1973). Environment and exposure to solvent of protein atoms. Lysozyme and insulin. Journal of Molecular Biology, 79(2), 351–371. https://doi.org/10.1016/0022-2836(73)90011-9

Smith, C. A., & Kortemme, T. (2008). Backrub-like backbone simulation recapitulates natural protein conformational variability and improves mutant side-chain prediction. Journal of Molecular Biology, 380(4), 742–756. https://doi.org/10.1016/j.jmb.2008.05.023

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression.

Song, Y., Tyka, M., Leaver-Fay, A., Thompson, J., & Baker, D. (2011). Structure-guided forcefield optimization. Proteins, 79(6), 1898–1909. https://doi.org/10.1002/prot.23013

Syed, A. A., Md., I. H., Asimul, I., & Faizan, A. (2014). A Review of Methods Available to Estimate Solvent-Accessible Surface Areas of Soluble Proteins in the Folded and Unfolded States. Current Protein & Peptide Science, 15(5), 456–476.

Talevi, A. (2018). Computer-Aided Drug Design: An Overview. In M. Gore & U. B. Jagtap (Eds.), Computational Drug Discovery and Design (pp. 1–19). Springer. https://doi.org/10.1007/978-1-4939-7756-7_1

Thiltgen, G., & Goldstein, R. A. (2012). Assessing Predictors of Changes in Protein Stability upon Mutation Using Self-Consistency. PLoS ONE, 7(10). https://doi.org/10.1371/journal.pone.0046084

Viricel, C., de Givry, S., Schiex, T., & Barbe, S. (2018). Cost function network-based design of protein-protein interactions: Predicting changes in binding affinity. Bioinformatics (Oxford, England), 34(15), 2581–2589. https://doi.org/10.1093/bioinformatics/bty092

Vivcharuk, V., Baardsnes, J., Deprez, C., Sulea, T., Jaramillo, M., Corbeil, C. R., Mullick, A., Magoon, J., Marcil, A., Durocher, Y., O'Connor-McCourt, M. D., & Purisima, E. O. (2017). Assisted Design of Antibody and Protein Therapeutics (ADAPT). PLOS ONE, 12(7), e0181490. https://doi.org/10.1371/journal.pone.0181490

Webb, B., & Sali, A. (2016). Comparative Protein Structure Modeling Using MODELLER. Current Protocols in Bioinformatics, 54, 5.6.1-5.6.37. https://doi.org/10.1002/cpbi.3

Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

Wu, T. T., & Kabat, E. A. (1970). An analysis of the sequences of the variable regions of Bence Jones proteins and myeloma light chains and their implications for antibody complementarity. The Journal of Experimental Medicine, 132(2), 211–250.

Xiong, P., Zhang, C., Zheng, W., & Zhang, Y. (2017). BindProfX: Assessing Mutation-Induced Binding Affinity Change by Protein Interface Profiles with Pseudo-Counts. Journal of Molecular Biology, 429(3), 426–434. https://doi.org/10.1016/j.jmb.2016.11.022

Xiong, Y., Wang, Q., Yang, J., Zhu, X., & Wei, D.-Q. (2018). PredT4SE-Stack: Prediction of Bacterial Type IV Secreted Effectors From Protein Sequences Using a Stacked Ensemble Method. Frontiers in Microbiology, 9. https://doi.org/10.3389/fmicb.2018.02571

Yin, R., Tran, V. H., Zhou, X., Zheng, J., & Kwoh, C. K. (2018). Predicting antigenic variants of H1N1 influenza virus based on epidemics and pandemics using a stacking model. PLOS ONE, 13(12), e0207777. https://doi.org/10.1371/journal.pone.0207777

9 ANEXOS

9.1 Instalación de ABPRED

ABPRED es distribuido como un programa de interfaz de línea de comandos desarrollado en lenguaje de programación Python v3.7 o superior, y por ahora, solo para sistemas Linux y macOSX. Para instalar Python y ABPRED recomendamos usar Conda (https://docs.conda.io/en/latest/), el cual puede ser instalado con *Anaconda* o *Miniconda* (https://docs.conda.io/en/latest/miniconda.html). Puedes seguir las instrucciones de instalación dependiendo de tu sistema en el link anterior.

Una vez instalado *Conda* procedemos a instalar ABPRED. Para ello primero clonaremos el repositorio de ABPRED:

```
$ git clone https://github.com/DatagenCL/AbPred
```

Esto descargara el proyecto en el directorio de trabajo, por lo cual es importante ejecutarlo en el directorio deseado desde la terminal.

Finalmente, la forma más sencilla de instalar el proyecto es usar Conda junto al archivo `environment.yml`, el cual está configurado para crear automáticamente un ambiente virtual llamado `abpred-dev` junto a todas las librerías necesarias

para su funcionamiento. Una vez creado el ambiente, activamos el ambiente e instalamos ABPRED en modo desarrollador usando *pip*:

```
$ cd abpred
$ conda env create -f environment.yml
$ conda activate abpred-dev
$ pip install -e .
```

Como último paso es necesario instalar programas de terceros, como FoldX y POPS. Las instrucciones para descargar e instalar POPS están en https://github.com/Fraternalilab/POPScomp/wiki/Installation. Para adquirir FoldX se debe solicitar una licencia académica o comercial en http://foldxsuite.crg.eu/. Finalmente, una vez instalado ambos programas solo queda modificar el archivo `conf.py` dentro de la acarpeta `abpred` y especificar la ruta absoluta de los ejecutable para FoldX y POPS (variables FOLDX_DIR y POPS, respectivamente).

9.2 Uso de ABPRED

Una vez instalado ABPRED, puedes ejecutar el comando `abpred run -h` en la terminal y obtendrás algo como esto:

```
usage: abpred run [-h] [--structure_file STRUCTURE_FILE] [-m MUTATIONS]
                  [-t RUN_TYPE] --partners PARTNERS [--models_dir
MODELS_DIR]
```

```
                [-v]

    abpred
    Antibody-Antigen affinity change prediction upon single mutations
    using stacking learning models

    Usage:
        Affinity change prediction (run mode)
        abpred run -p <structure file> -m <mutation or mutation_list> --partners <interacting chains>
        $ abpred run -p 3bdy.pdb -m mutants_list.txt --partners HL_V

    Requeriments:
        TODO..


optional arguments:
  -h, --help            show this help message and exit
  --structure_file STRUCTURE_FILE, -p STRUCTURE_FILE
                        Full filename (including path) of the PDB file that you wish to mutate
  -m MUTATIONS, --mutations MUTATIONS

                        Mutation(s) that you wish to evaluate.

                        A Mutation List File. Provide one mutation per line as <pdb_chain> <mutation>.
                        (e.g   H R282W
                               H Y236F
                               H N239Y
                               H C242S)
  -t RUN_TYPE, --run_type RUN_TYPE
                        Type of analysis to perform. Must be one of:
                        0 | all:
                            Build and Repair models, Calculate Features and predict mutations.
                        1 | model:
                            Build models and Repair.
                        2 | model.features
                            Build models, Repair and Calculate Features
                        3 | features.predict:
```

```
                          Calculates features and predict mutations
(assumes Models has been calculated previously)
                      4 | predict:
                          Predict binding energy change based on
features (assumes features data has been calculated)
  --partners PARTNERS
                      PDB structure chains to be considered as
partners for calculations.

                      Partners must be provided as
<partnerA>_<partnerB>. You can specify more than
                      one chain per partner.
                      (e.g. 'A_B, HL_A, HL_ABCD')
  --models_dir MODELS_DIR
                      Folder containing precalculated Structural
models. Models of wildtype and mutant need
                      to have .wt.pdb and .mut.pdb tags respectively.
  -v, --verbose       Increase verbosity level. Can be specified
multiple times
```

Puedes ir a la carpeta `tests` y ejecutar un ejemplo típico de prueba. Para predecir el cambio de energía sobre un complejo Ag-Ab y una lista de mutaciones puntuales, ABPRED requiere la estructura PDB del complejo, un archivo de texto con las mutaciones puntuales. Una mutante por línea en formato CADENA_TABULACION_MUTANTE (ej ‘H D98K’). Por último, requiere las cadenas interactuantes o *partners,* en formato PARTNERA_PARTNERB:

```
$ abpred run -p 3bdy.pdb -m adapt_mutants.txt --partners HL_V -v
```