UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA QUÍMICA,
BIOTECNOLOGÍA Y MATERIALES

APLICACIONES DE MACHINE LEARNING Y DATA MINING EN INGENIERÍA DE PROTEÍNAS: DISEÑO E IMPLEMENTACIÓN DE NUEVAS ESTRATEGIAS PARA EL ESTUDIO DE MUTACIONES

TESIS PARA OPTAR AL GRADO DE
DOCTOR EN CIENCIAS DE LA INGENIERÍA, MENCIÓN INGENIERÍA QUÍMICA Y BIOTECNOLOGÍA

DAVID ALFREDO MEDINA ORTIZ

PROFESOR GUÍA:
DR. ÁLVARO OLIVERA NAPPA

PROFESOR CO-GUÍA:
DR. JUAN A. ASENJO DE LEUZE DE LANCIZOLLE

MIEMBROS DE LA COMISIÓN:
DR. GONZALO NAVARRO BADINO
DR. BÁRBARA ANDREWS FARROW
DR. MAURICIO MARÍN CAIHUÁN

SANTIAGO DE CHILE
2022

Resumen

La incorporación de las técnicas de aprendizaje de máquinas y minería de datos a las estrategias de diseño de mutaciones ha permitido mejorar enormemente su rendimiento y sus facilidades de aplicación. No obstante, diversos desafíos aparecen al agregar estas metodologías computacionales en los protocolos de diseños experimentales.

Debido a las variadas problemáticas existentes, esta tesis de doctorado se ha centrado en el diseño e implementación de metodologías computacionales que permitan solventar los desafíos de la incorporación de las técnicas de aprendizaje de máquinas a los protocolos de diseño de proteínas actuales, proponiendo la elaboración de una plataforma de manejo de datos para ingeniería, la cual mejora el rendimiento de modelos predictivos para variadas tareas y permite el diseño de mutaciones con propiedades deseables, contribuyendo en diferentes aristas de desarrollo.

Primero, se diseñó e implementó una estrategia de representación numérica de secuencias de proteínas combinando codificadores basados en propiedades fisicoquímicas semánticamente seleccionados con transformaciones de Fourier, con el fin de mejorar el proceso de codificación para aplicaciones de algoritmos basados en técnicas de *machine learning*. En una segunda etapa, se diseñó e implementó un *framework* de entrenamiento de modelos predictivos para tareas de ingeniería que proteínas. Este sistema emplea la estrategia de representación numérica propuesta en este trabajo de doctorado como input para entrenar modelos basados en algoritmos de aprendizaje supervisado, los cuales se optimizan su rendimiento mediante la selección eficiente de hiperparámetros mediante algoritmos genéticos. Estos métodos se combinan en un único sistema por medio de sistemas de aprendizaje ensamblado para desarrollar el sistema predictivo de interés. Finalmente, se diseñaron e implementaron estrategias de diseño de proteínas mediante la elaboración de metodologías para explorar espacios latentes y reconstrucción de *landscapes*. Además, se construyó una estrategia de identificación de sitios relevantes en proteínas, combinando los puntos de vista filogenéticos, termodinámico y estructural, con el fin de favorecer las herramientas de diseño de mutaciones sitio dirigidas y el análisis de trayectorias en procesos evolutivos.

Todas las metodologías diseñadas e implementadas en este trabajo se validaron con diferentes conjuntos de datos habilitados en la literatura y se compararon con estrategias previamente reportadas, logrando, en la mayoría de los casos, obtener mejores rendimientos en cuanto a calidad de predicciones, así como también facilidades en interpretación de los resultados, gracias al tipo de algoritmos empleados, siendo directamente relacionado con los ideales de la *Inteligencia Artificial Explicable*, lo cual denota la relevancia de las metodologías planteadas para los campos de ingeniería de proteínas y biotecnología.

A mi familia, amigos, profesores, colegas y a todos los que hicieron esto posible ...

Agradecimientos

La lista de agradecimientos es enorme y variada. Pero... tratemos de ordenarla!

En el aspecto académico, quiero agradecer al profesor Álvaro Olivera, quien ha sido mi mentor y una de mis inspiraciones a seguir desde que ingresé a la universidad, su apoyo ha sido fundamental y su guía durante este proceso ha sido invaluable, sin lugar a duda, el tipo de científico que me he transformado es gracias a él. En este mismo ámbito agradezco a los profesores Carlos Conca y Juan Asenjo, quienes siempre han confiado en mí y me han apoyado en las diversas postulaciones tanto a becas como pasantías, brindando confianza y sobre todo, su apoyo. A los profesores Frances H. Arnold, Mehdi D. Davari, Roberto Uribe y Marcelo Navarrete, por confiar en mí y darme la oportunidad de colaborar y trabajar con ellos en sus laboratorios y proyectos. En este ámbito, me gustaría agradecer a CeBiB, por su apoyo y auspicio durante este tiempo, lo cual significó mucho para mí y en variadas ocasiones me sacaron de apuro. Finalmente, agradezco a ANID por financiar mi doctorado gracias a la beca de doctorado nacional.

En el ámbito personal, agradezco a Sebastián Contreras, mi partner académico, quien no solo me apoya en ideas locas y extravagantes, sino que también ha demostrado su apoyo en todo ámbito de mi formación. Gracias por llegar a trabajar en el proyecto OGTT y reemplazar al que no debe ser nombrado! hahaha.

Por otro lado, agradezco a Cristian y Catalina, por darme la oportunidad de trabajar con ellos y conocer el mundo privado, brindarme conocimiento y creer en mí para trabajos completamente diferentes a mi área, otorgándome buenos momentos con su compañía. Por supuesto, también a mis amigos Gonzalo, Leandro y Diego, quienes siempre me han dado ánimos y motivado a seguir, su amistad desde la universidad aún persiste y se han transformado en pilares de mi vida. Además, agradezco a Gabriel quien solo llego por adquirir conocimiento al grupo y ahora es prácticamente mi mano derecha desde el componente bioinformático, si hay alguien en quien me puedo apoyar, es en él.

Por último, a mi familia, quienes son el pilar de mi vida, me cuidan siempre y guían mi camino a pesar de la distancia. La persona que soy ahora es gracias a ellos y siempre estarán en mi mente y corazón, me siento orgulloso de ser su hijo y familiar!

A todos... Mil gracias a todos!

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción y contexto general

Diseñar proteínas con propiedades deseables es uno de los principales desafíos de la ingeniería de proteínas de las últimas décadas. Los métodos como la evolución dirigida y el diseño racional han sido las principales técnicas empleadas para este propósito. Sin embargo, pese a sus significantes aportes, no funcionan de manera general, necesitan de maquinarias especializadas o de conocimiento específico para poder aplicarse con éxito, siendo necesario el diseño e implementación de herramientas computacionales que permitan apoyar estas técnicas experimentales [174].

Los métodos computacionales basados en estrategias bioinformáticas han permitido estudiar los efectos de mutaciones en las proteínas. Variadas aplicaciones se han centrado en el estudio de la estabilidad térmica de las proteínas y cómo dichas mutaciones provocan cambios termodinámicos, los cuales pueden alterar su estabilidad, siendo las más relevantes *Site Directed Mutator* (SDM) [121] y *FOLD-X* [148]. Por otro lado, existen herramientas basadas en el uso de propiedades filogenéticas y técnicas estadísticas para clasificar las mutaciones como benéficas o dañinas con base en la conservación ancestral de zonas en las proteínas, siendo ejemplos de este tipo la librería *MOSST* [118] y la herramienta *EvCoupling* [162]. Un punto importante a destacar es que, a la fecha, no existen herramientas o metodologías para el estudio de mutaciones que combinen ambos puntos de vista y demuestren cuáles son los principales focos de interés para estudiar mutaciones, siendo un posible campo de interés para teorías de información, *machine learning* y aplicaciones bioinformáticas.

Recientemente, los métodos basados en *machine learning* han demostrado su utilidad en ingeniería de proteínas, siendo el mayor avance a la fecha el diseño e implementación de *AlphaFold* [80], herramienta que predice de manera exitosa y con alta precisión la estructura secundaria de las proteínas desde su información lineal. Por otro lado, variados métodos predictivos han sido desarrollados con éxito para tareas particulares de ingeniería de proteínas, tales como, la clasificación funcional de enzimas, la identificación de proteínas que interaccionan con DNA, o la predicción de actividad biológica en secuencias de péptidos, entre un gran número de aplicaciones [107]. Sin embargo, dichos métodos, en su mayoría, carecen de generalización, no demuestran una transferencia de aprendizaje eficiente y están limitados al conjunto de entrenamiento empleado. A pesar de ello, estos métodos han sido incorporados a los protocolos de diseño de proteínas, naciendo las áreas de *machine learning directed*

evolution [174, 181] y el diseño semi racional de proteínas[153].

Pese a los grandes esfuerzos tanto en desarrollo computacional como de mejoramiento de las técnicas experimentales, diseñar proteínas con propiedades deseables, predecir la funcionalidad o actividad biológica, estudiar los cambios en los plegamientos, evaluar los efectos de las mutaciones, entre otros, son problemas que aún persisten, siendo la motivación principal del desarrollo de esta tesis de doctorado, aportando a la comunidad científica con el diseño e implementación de métodos, estrategias y herramientas para el área de biología computacional con aplicación directa a los campos de investigación previamente mencionados y con especial énfasis en el área de la biotecnología y la ingeniería de proteínas.

Con el fin de facilitar la comprensión del documento expuesto, de manera general, esta tesis de doctorado se divide en seis capítulos, los cuales abordan el estado de arte, diferentes problemáticas específicas asociadas al objetivo general y a la hipótesis planteada de este proyecto, así como también, las diferentes conclusiones y proyecciones del trabajo planteado. A pesar de que cada capítulo resuelve un problema en específico con su metodología particular, son dependientes entre sí, ya que juntos abordan el problema de diseñar y explorar mutaciones con propiedades deseables. Estos capítulos se resumen a continuación.

1. **Capítulo 1, Introducción y contexto general**: En este capítulo se exponen los conceptos generales de ingeniería de proteínas, técnicas de diseño, herramientas computacionales, y *machine learning*, entre otros. Además, se hace énfasis en el estado de arte actual y las variadas problemáticas y desafíos existentes en la ingeniería de proteínas.

2. **Capítulo 2, Representaciones numéricas**: Contempla el diseño e implementación de estrategias de representación numérica de secuencias de proteínas, para facilitar el entrenamiento de modelos predictivos empleando algoritmos de *machine learning* clásico.

3. **Capítulo 3, Modelos predictivos ensamblados**: Abarca el diseño e implementación de estrategias de mejoramiento de rendimiento de modelos predictivos y la sinergia entre la representación numérica de las secuencias desde diferentes puntos de vista con variados algoritmos de *machine learning* clásico.

4. **Capítulo 4, Estrategias de diseño de mutaciones y exploración de espacios latentes**: Se diseña e implementa una estrategia de reconstrucción de *landscapes* para el diseño de secuencias con propiedades deseables, combinando modelos predictivos basados en *machine learning* y modelos estadísticos, siendo testeado en diseño de péptidos con actividad anti-HIV y sistemas de predicción de productividad en sistema de producción recombinante.

5. **Capítulo 5, Estudio de mutaciones puntuales**: Se diseña e implementa una metodología para identificar sitios de interés en proteínas a partir de combinación de propiedades filogenéticas, termodinámicas, estructurales, y el uso de modelos epistáticos.

6. **Capítulo 6, Conclusiones, trabajos en desarrollo y perspectivas a futuro**: Contempla las principales conclusiones de las metodologías propuestas, los trabajos en desarrollo y las posibles vías de continuación y contribución científica.

A continuación, se exponen los diferentes conceptos de ingeniería de proteínas y del *machine learning*. Además, se describe el estado de arte reciente y los más relevantes avances, identificando las principales falencias y problemáticas, así como también, los desafíos latentes en el área, los cuales son focos de inspiración para las metodologías desarrolladas en este trabajo de doctorado.

1.1. Ingeniería de proteínas y estrategias de diseño de variantes

La ingeniería de proteínas es un campo de investigación centrado en el diseño y generación de proteínas útiles o con propiedades deseables, con un enfoque principal en la comprensión del plegamiento de las proteínas [95].

Actualmente, existen dos estrategias principales para el diseño de proteínas, siendo estas la evolución dirigida y el diseño racional. A pesar de ser dos técnicas diferentes, existen variadas situaciones donde se pueden combinar para obtener mejores resultados. Importante mencionar que, un estudio completo de residuos y una evaluación detallada de las sustituciones es una limitante importante para estas dos técnicas, debido a recursos económicos, humanos y de capacidad técnica, entre otros [95].

La evolución dirigida imita el proceso de selección natural, permitiendo direccionarla hacia objetivos definidos, reflejados en cuanto a funciones o propiedades deseables [99, 10]. Una representación del proceso se observa en la Figura 1.1.

El proceso, de manera general, consiste en someter un gen de interés a rondas iterativas de mutagénesis, con el fin de crear una biblioteca de variantes. A partir de dicho conjunto de elementos, se seleccionan las variantes con la función deseada. Finalmente, se aíslan y amplifican para formar una plantilla para la siguiente iteración. Así, el proceso sigue iterando y estadísticamente se seleccionan las más favorables y aquellas que tendieron a la evolución debido a la supervivencia en el proceso [10].

Con respecto al diseño racional de proteínas, esta es una técnica ampliamente utilizada y al igual que la evolución dirigida, tiene el objetivo general de generar variantes con alguna función de interés o características particulares. No obstante, exhibe una diferencia relevante, la cual se centra en la información que debe existir sobre la estructura, mecanismos, plegamiento o secuencia lineal de la proteína de interés [26], complicando su desarrollo en etapas iniciales del diseño de variantes para proteínas sin estructura o comprensión de sus mecanismos generales.

En la actualidad, ambos métodos se benefician de técnicas computacionales para su desarrollo, facilitando su aplicación e incrementando el conocimiento sobre los sistemas de interés, emergiendo las estrategias de *machine learning directed evolution* [174] y diseño semi racional [153].

El *machine learning directed evolution* combina las técnicas de aprendizaje supervisado con la evolución dirigida para generar la selección de secuencias guiada por los modelos

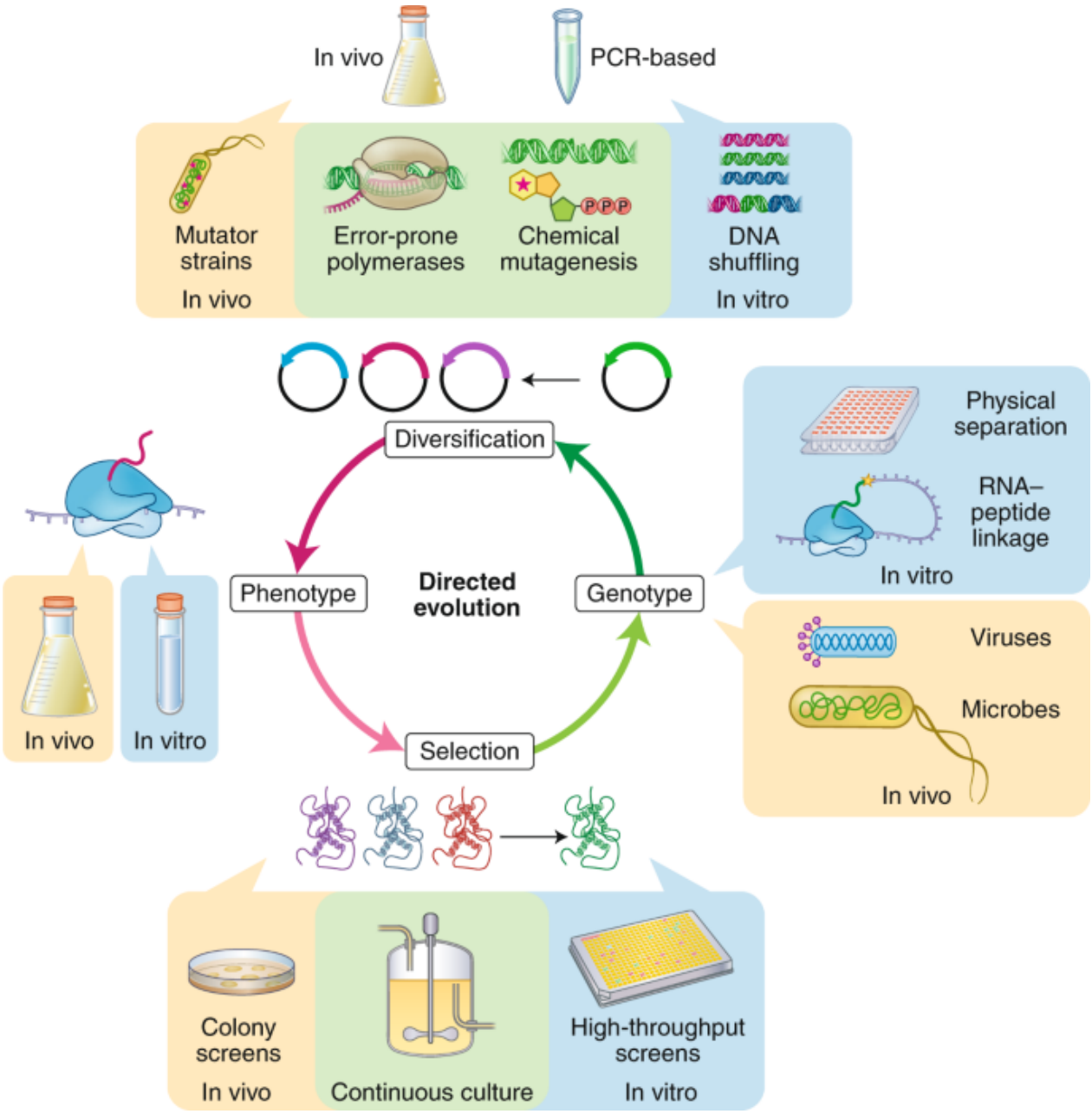


Figura 1.1: Esquema representativo de evolución dirigida, extraído desde [111]

predictivos. La Figura 1.2 muestra las diferentes evoluciones que ha sufrido el proceso de evolución dirigida desde su etapa inicial (Figura 1.2 A), etapa de combinaciones aleatorias (Figura 1.2 B) y la incorporación de métodos predictivos basados en *machine learning* (Figura 1.2 C) a través del cual se guía la evolución dirigida.

De manera más general, todos los métodos experimentales que se asocien con métodos computacionales para apoyar el diseño de variantes, comprenden los sistemas basados en diseño semi racional [153, 157].

Con base en lo anterior, las técnicas computacionales y las estrategias de *machine learning* y *deep learning* han adquirido un rol fundamental en beneficio de los métodos de diseño de

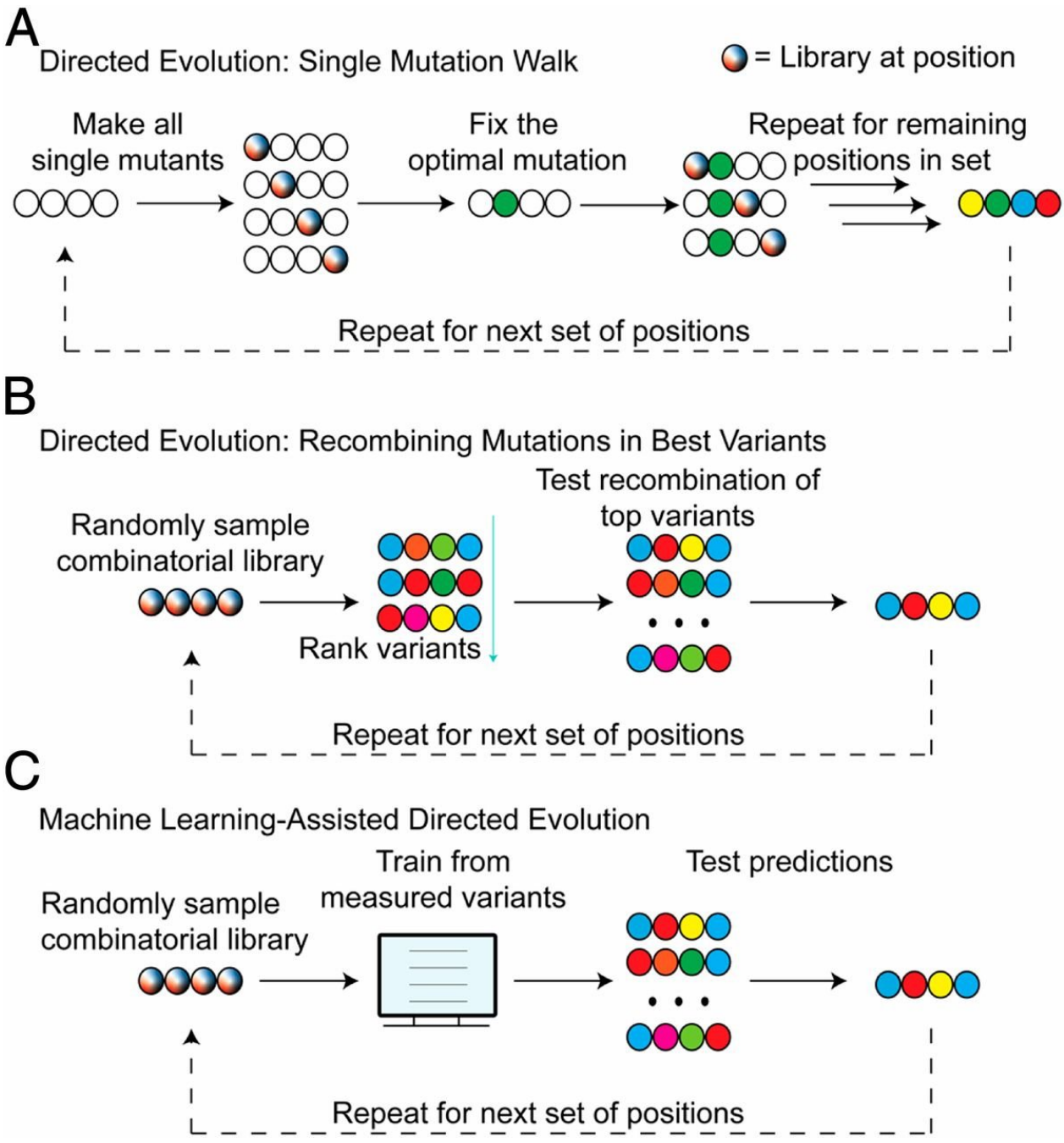


Figura 1.2: Desde la evolución dirigida hasta el *machine learning directed evolution*, extraída desde [177]

proteínas. A continuación, se resumen de manera general las principales herramientas y el estado de arte actual y más relevante de las técnicas computacionales aplicadas a ingeniería de proteínas.

1.2. Métodos computacionales aplicados en ingeniería de proteínas

Diferentes métodos computacionales han sido desarrollados para distintos análisis, con el fin de poder responder variadas interrogantes planteadas desde enfoques diversos, ya sea, para estudiar secuencias lineales, filogenia y motivos conservados, análisis de estructuras, modelamiento estructural, estudio de mutaciones, entre otros. A continuación, se describen algunos de los principales enfoques y las herramientas existentes para su desarrollo.

1.2.1. Métodos de análisis filogenéticos

Los análisis filogenéticos se centran en el estudio de secuencias lineales de proteínas o genes, con el fin de identificar parentesco, motivos conservados o reconocimiento de dominios. Las principales metodologías se basan en realizar alineamientos de secuencia para reconocer identidades de la secuencia de estudio con respecto a información reportada previamente.

Una de las herramientas más conocidas para el desarrollo de alineamientos de secuencias es *Blast* [83], la cual permite hacer múltiples comparaciones de secuencias de interés versus bases de datos con genes o proteínas, empleando algoritmos de alineamiento local [7].

Por otro lado, se encuentran los alineamientos múltiples de secuencias, los cuales son utilizados en la comparación de secuencias lineales con el fin de encontrar patrones o motivos conservados, o, la identificación de parentesco en grupos de familias [163]. Esto último, es bastante empleado cuando se analizan secuencias de organismos no reportados y cuya función se desea identificar, siendo uno de los ejemplos más recurrentes, la anotación de genomas [51]. Una de las herramientas más utilizadas en esta área es el software *MegaX* y los algoritmos como *ClustalW* [66].

A su vez, el uso de los conceptos filogenéticos, ha sido empleado para el estudio de propensiones de mutaciones y cómo estas afectan a la función de la proteína. Considerando esto, se han diseñado e implementado herramientas como *MOSST* [118] y *Evcouplings* [73], las cuales han permitido comprender la propensión de residuos y posiciones relevantes en secuencias desde el punto de vista filogenético, solo estudiando conjuntos de elementos a nivel probabilístico, sin la necesidad de conocer estructuras tridimensionales de las proteínas. No obstante, estos análisis requieren normalmente de secuencias con un grado de divergencia, sin perder el concepto de *Familia*, siendo necesarios métodos de automatización para la generación de estas. Además, en el caso de *MOSST*, se requiere representar numéricamente las secuencias a partir de propiedades fisicoquímicas, con el fin de poder llevar a cabo los análisis estadísticos de distribución de propiedades.

Finalmente, una de las herramientas más interesantes del último tiempo es *EVcouplings* [73], la cual se basa en estrategias de *evolutionary coupling*, combinadas con modelos de grafos por pares no dirigidos a partir de múltiples alineamientos de secuencia, con el fin de desarrollar modelos epistáticos y probabilísticos independientes para facilitar la identificación de sitios epistáticos y la relevancia de los efectos de mutaciones desde un punto de vista beneficioso

o dañino. No obstante, presenta los mismos problemas expuestos para las herramientas de estudios filogenéticos, así como también, un enorme costo computacional debido al desarrollo de los modelos de grafos.

Estos estudios generan las bases fundamentales para el análisis de secuencias y su principal y más relevante característica es que solo se necesita la secuencia lineal a estudiar y a partir de ella, es factible comprender un panorama relacionado con patrones de conservación, tendencias, relaciones evolutivas o inclusive, propensiones y posiciones relevantes. No obstante, no son los únicos, ya que existen herramientas computacionales que facilitan la predicción de la estructura secundaria, funcionalidad, entre otros, siendo una de las principales herramientas *Swiss-Prot* [17].

1.2.2. Métodos de análisis de estructuras

Los métodos de análisis de estructuras, tienen el objetivo de comprender patrones de interacción, efectos de energía y estudiar diferentes propiedades fisicoquímicas y termodinámicas, a partir de la estructura tridimensional de una proteína, la cual puede ser obtenida por cristalografía de rayos X o por medio de resonancia magnética nuclear, entre los principales métodos de obtención.

No obstante, también es factible el desarrollo de modelos estructurales de proteínas a partir de secuencias lineales, ya sea mediante modelamiento por homología o por técnicas de *ab-initio*. Diferentes software permiten la implementación de estas técnicas, dentro de los cuales se encuentran *SWISS-MODEL* [65], *IntFOLD* [106], *ROSETTA* [90], y *MODELLER* [54], entre los principales.

Por otro lado, existen diferentes métodos computacionales que facilitan el estudio de interacción entre proteína y una molécula o proteína-proteína, los cuales principalmente se enfocan en el uso de técnicas como *docking* o dinámicas moleculares, con el fin de estudiar los posibles residuos que participan en la interacción, evaluándose a nivel energético y midiendo el desempeño en términos de medidas de error [27]. A su vez, técnicas basadas en simulaciones moleculares, permiten comprender la interacción en sí y emular el comportamiento entre la molécula de interés y la proteína. Además, métodos computacionales basados en química cuántica, han sido utilizados para comprender fenómenos de interacción a una escala mucho más precisa. No obstante, estos son ampliamente más costosos computacionalmente y su uso es limitado al estudio de un número pequeño de átomos [12].

Existen diferentes herramientas que permiten hacer dinámicas moleculares, tales como *NAMD* [127] y *AMBER* [28], mientras que para la interacción entre moléculas, o *docking*, existen herramientas como *AutoDock* [165], *RosettaDock* [100], y *GRAMM-X* [164], entre otras. Además, se encuentra la suite Maestro *Shrödinger* [140], la cual abarca funcionalidades para las diferentes acciones propuestas.

Todas las herramientas computacionales nombradas se centran en el estudio de las estructuras, razón por la cual dependen no solo de la existencia del cristal, sino que también, los resultados generados están estrechamente relacionados con la calidad de la estructura o del modelo tridimensional brindado como input. Pese a ello, sigue siendo uno de los focos de

mayor interés, ya sea para comprender los mecanismos de interacción y sus aplicaciones en campos como la medicina, o analizar mecanismos de funcionamiento para diseño de variantes. Dado esto, los esfuerzos en bioinformática estructural se han centrado principalmente en la disminución del costo computacional, basados principalmente en emplear estrategias de paralelización y distribución de cálculos, con el fin de disminuir la brecha entre calidad y tiempo de ejecución.

1.2.3. Métodos de estudio de mutaciones

De modo general, los estudios de mutaciones se basan principalmente en el análisis de la estructura ante los cambios de residuos o la adición o eliminación de estos, evaluando las variaciones mediante diferencias de energía libre, entre la proteína inicial y la mutada. Herramientas como *FoldX* [149], *SDM* [122], *Auto-Mute* [104], entre otras, permiten analizar cómo afecta una mutación en términos energéticos, basándose para ello, en el uso de funciones de energía potencial y dinámicas moleculares asociadas a dicha sustitución. Sin embargo, el uso de este tipo de herramientas, conlleva un gran costo computacional debido a los diferentes cálculos que son requeridos. Por otro lado, herramientas como *MOSST* y *EvCoupling* permiten evaluar los efectos de las mutaciones a nivel filogenético. No obstante, solo pueden clasificar las mutaciones desde el punto de vista benéfico en términos relevantes al alineamiento de secuencias, dejando de lado como estas variantes afectan a la estructura y por ende a la estabilidad de la proteína. Además, todas las herramientas, metodologías y estrategias nombradas durante esta sección no consideran técnicas basadas en *machine learning* y *data mining*, ya que, serán consideradas durante las siguientes secciones de este capítulo, así como también durante todo el desarrollo de esta tesis de doctorado.

1.3. Inteligencia artificial y machine learning

La inteligencia artificial es un conjunto de técnicas que se asocian a la generación de conocimientos y el traspaso de aprendizaje entre distintos sistemas de interés, siendo su principal enfoque, el diseño e implementación de metodologías para entregar autonomía a sistemas complejos [71]. Dentro de estas técnicas se encuentra el *machine learning*, encargándose de generalizar los comportamientos para aprender de los patrones, con el fin de predecir nuevas acciones para fuentes desconocidas [115]. Sus aplicaciones en ingeniería de proteínas han sido diversas, logrando desarrollar modelos predictivos para clasificación de funciones de enzimas [105], clasificación de proteínas de unión a DNA [129], evaluación de actividad biológica de péptidos [130], predicción de efectos de mutaciones sobre la estabilidad, enantioselectivad y variadas propiedades fisicoquímicas [182, 107], entre otras, siendo la lista tan extensa como propiedades se desee analizar combinadas con investigadores generando nuevas estrategias.

Uno de los logros más relevantes del *machine learning* a la fecha, ha sido, *Alpha-Fold* [80] herramienta computacional que logra predecir las estructuras secundarias de las proteínas empleando estrategias de *deep learning*, siendo ampliamente aplicada en diferentes casos de estudio [6]. La Figura 1.3 muestra un esquema representativo de la arquitectura generada para entrenar *AlphaFold*, donde se contempla la implementación de alineamientos de secuencia

múltiples para analizar los parentescos de las secuencias, además de la incorporación de análisis de aristas entre secuencias y residuos así como también entre residuos, con el fin de poder armar matrices de distancia y mapas de contacto que facilitan el desarrollo del plegamiento estructural de la proteína de interés.

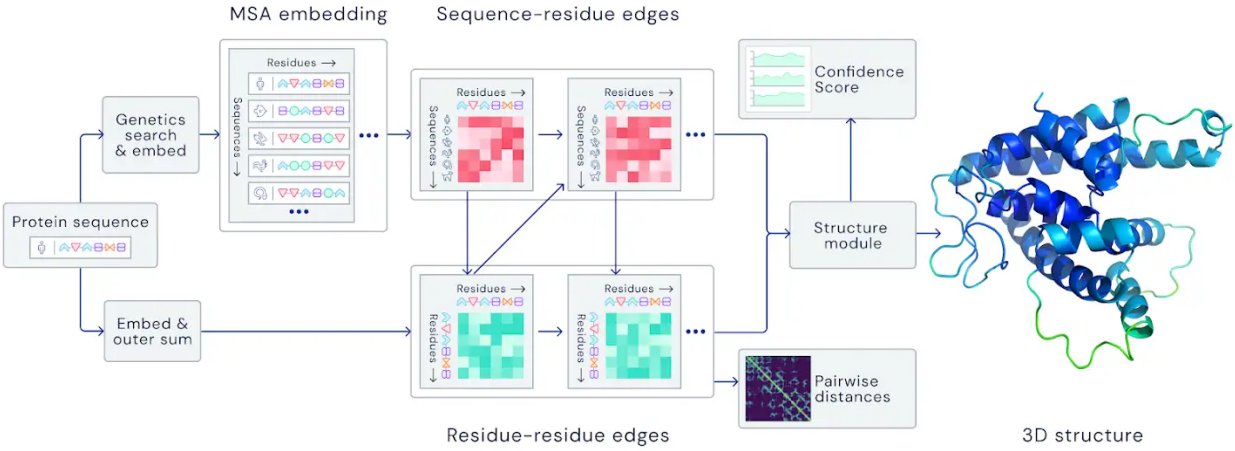


Figura 1.3: Esquema representativo de arquitectura diseñada para entrenamiento de *AlphaFold*, extraída desde [80]

Las técnicas de diseño de proteínas se han visto beneficiadas con el uso del *machine learning*, naciendo los métodos basados en el diseño semi racional [99]. A su vez, ha nacido el *machine learning directed evolution*, facilitando la simulación de variantes sin emplear métodos experimentales [181, 174]. No obstante, a pesar de las diferentes aplicaciones y de los variados usos del ML en la ingeniería de proteínas, problemas como la generalización de comportamientos, predicción de funciones biológicas y estimaciones de reacciones enzimáticas aún persisten, siendo de vital importancia el desarrollo de nuevas metodologías y estrategias computacionales para su solución.

Debido a que todo el desarrollo de la tesis se centrará en la aplicación del *machine learning* y las diferentes técnicas de aprendizaje. A continuación, y con el fin de facilitar la comprensión de las diferentes temáticas a abordar durante esta tesis de doctorado, se detallarán de manera breve los principales conceptos, sus definiciones, los algoritmos asociados y los puntos importantes para desarrollar modelos predictivos, aplicando *machine learning*. Además, se abarcarán los conceptos de redes neuronales y *deep learning* con el fin de ampliar el *background* de estas temáticas.

1.4. Machine learning y tipos de aprendizaje

El *machine learning* se centra en la generalización de comportamientos con el fin de aprender una acción a partir de conocimiento previo. Actualmente, existen diferentes tipos de aprendizaje, los cuales se clasifican con respecto a las tareas a resolver, siendo estos resumidos en la Figura 1.4. Además, un resumen detallado de los tipos de aprendizaje se expone en la Tabla 1.1.

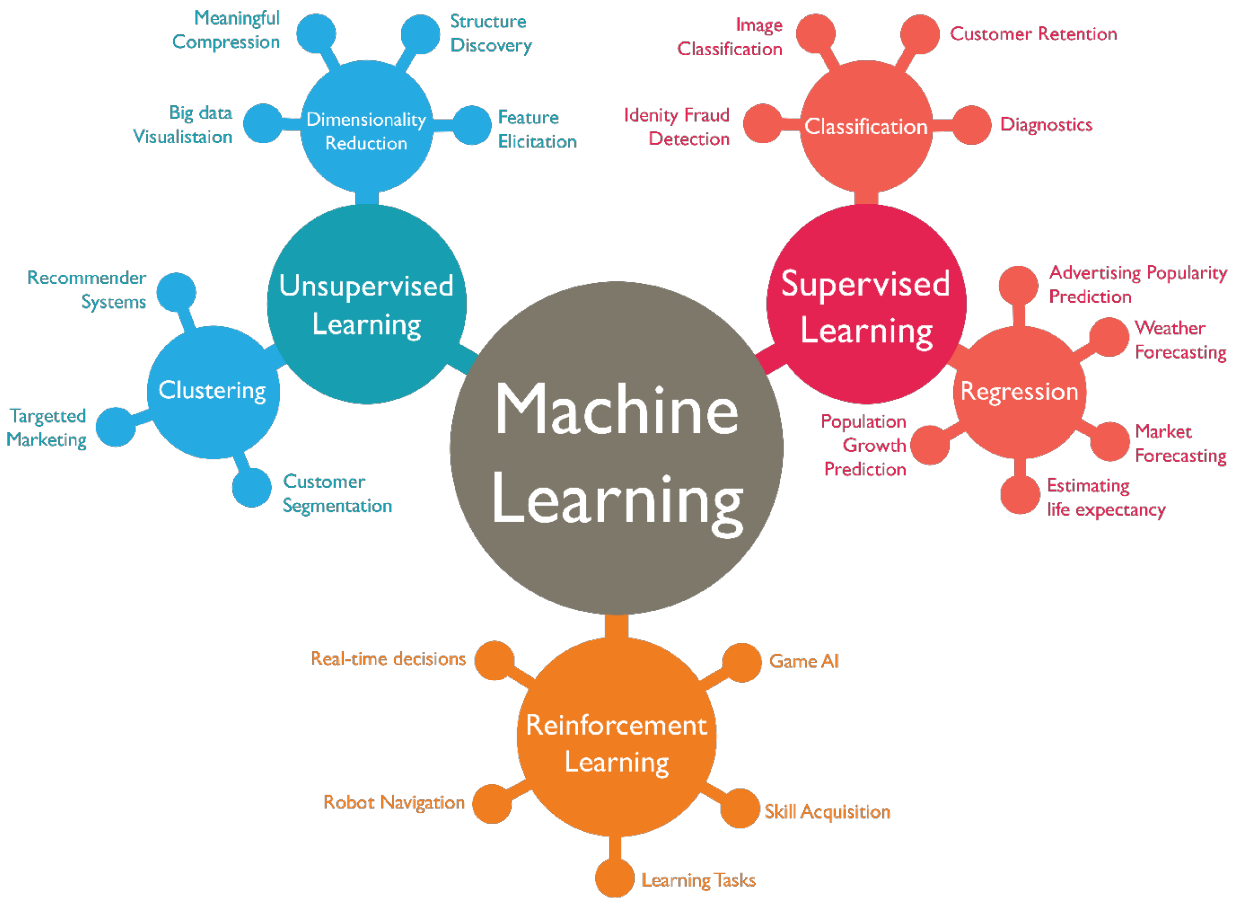


Figura 1.4: Resumen de los principales tipos de aprendizaje en el *machine learning*.

Tipo de aprendizaje	**Descripciones generales**	**Aplicaciones en ingeniería de proteínas**
Supervisado	Se centra en la predicción de nuevos ejemplos a partir del conocimiento previamente adquirido de conjuntos ya etiquetados.	Predicción de efecto de mutaciones en estabilidad Clasificación de funciones enzimáticas
No Supervisado	Se centra en la identificación de patrones en conjuntos de datos no etiquetados con el fin de generalizar estos comportamientos	Caracterización de grupos de variantes Identificación de patrones con efecto clínico
Reforzado	Se basa en penalizaciones y bonificaciones a medida que se generan acciones de aprendizaje con respecto a errores o aciertos de los modelos predictivos	Clasificación de actividad enzimática Predicción de reacciones enzimáticas
Semi-supervisado	Se centra en utilizar tanto datos etiquetados como no etiquetados para desarrollar modelos predictivos	Clasificación de mutaciones con efecto clínico para p53

Tabla 1.1: Resumen de los diferentes tipos de aprendizaje existentes en el *machine learning*.

A partir de lo expuesto en la Tabla 1.1 se desprende que dependiendo del problema a resolver, es posible emplear un tipo de aprendizaje en específico. Es decir, si se desea una identificación de patrones, se recomienda trabajar con aprendizaje no supervisado, en el caso datos etiquetados y con objetivo de desarrollar modelos predictivos se sugiere emplear aprendizaje supervisado. Para el caso de los desarrollos con aprendizaje semi-supervisado se recomienda trabajar con conjuntos de datos híbridos, pero en cuyos casos, los ejemplos no etiquetados sean similares a los etiquetados. Finalmente, en el caso de aprendizajes reforzados, se recomienda emplear cuando se actualizará el modelo con base en nuevos conocimientos, ya sea a raíz de actualización de datos experimentales o nuevos datos o información.

En esta tesis de doctorado, los algoritmos y las estrategias presentadas serán basándose en aprendizajes no supervisados y supervisados debido a la naturaleza de los datos procesados y el tipo de estrategias desarrolladas. Con base en lo anterior, se hablarán en más detalle de estos tipos de aprendizaje, contemplando algoritmos, evaluación de desempeño y diferentes problemas asociados a estas metodologías.

1.4.1. Aprendizaje supervisado

Tal como se mencionó previamente, el aprendizaje supervisado se centra en la predicción de nuevos ejemplos a partir del conocimiento existente, es decir, necesita conjuntos de datos etiquetados para poder generalizar los comportamientos asociados a la predicción y poder estimar nuevos valores. Tareas clásicas de este tipo de aprendizaje en ingeniería de proteínas son la clasificación de funciones, evaluación del efecto clínico de mutaciones y estimación de propiedades termodinámicas [174, 107]. Actualmente, existen dos principales focos de desarrollo, el *machine learning* clásico y el *deep learning*.

El *machine learning* clásico se centra en el uso de algoritmos de aprendizaje supervisado para cumplir tareas específicas. Ejemplos de estos algoritmos corresponden a árboles de decisión, *support vector machine*, *random forest*, entre otros. Por otro lado, el *deep learning* o aprendizaje profundo ha nacido para resolver problemas más complejos y con un gran volumen de datos, siendo empleado principalmente para tareas como predicción de complejos de interacción, desarrollo de modelos de estructura secundaria, entre otros, enfocándose en trabajar con objetos tales como imágenes, textos y estructuras de grafos [6].

Las principales diferencias entre ambos, se centran en la forma en cómo se representan los conjuntos de datos iniciales, así como también la interpretación de los resultados, la explicación de las predicciones y la evaluación del aprendizaje adquirido en cuanto a transparencia, siendo mucho más clara la interpretación para el *machine learning* clásico, debido a que los métodos de *deep learning* dado a su naturaleza de capas y a la emulación del funcionamiento cerebral, la forma en que se transmite la información y adquiere el conocimiento es no clara [109, 151].

1.4.2. Algoritmos clásicos de aprendizaje supervisado

Tal como se nombró previamente, los algoritmos clásicos de aprendizaje supervisado permiten una mayor comprensión del aprendizaje, así como también un mejor entendimiento e interpretación de las predicciones, principalmente debido a cómo funcionan. La Tabla 1.2 resume los principales algoritmos de aprendizaje supervisado, describiéndolos de manera resumida y mostrando algunas de sus ventajas y problemas que presentan a la hora de aplicarlos.

Tal como se observa en la Tabla 1.2 no se describen las redes neuronales, las cuales también corresponden a un algoritmo de aprendizaje supervisado. Sin embargo, los detalles de este tipo de algoritmo se describen a continuación.

Algoritmo	Descripción general
k-NN	Se basa en la predicción de nuevos ejemplos en base a la distancia que exista contra los ejemplos previamente etiquetados, desarrollando predicciones por cercanía y sistemas de votación. Es bastante intuitivo. Sin embargo, el poder computacional escala con respecto a los tamaños de los vectores a estimar la distancia. Además, emplear este método sólo en base a las distancias, resulta poco selectivo a la hora de clasificar ejemplos con similares características pero que con pequeñas perturbaciones su etiqueta cambie.
Árboles de Decisión	Representan conjuntos de reglas que permiten clasificar o predecir nuevos ejemplos. Las reglas se definen según la ganancia de información que brindan los atributos en base a funciones de entropía o entalpía. Representa uno de los algoritmos más fáciles de interpretar debido a que genera un árbol. No obstante, presenta problemas a la hora de trabajar con conjuntos de datos que no tienen diferencias significativas en sus atributos.
Support Vector Machine	Emplea estrategias de transformaciones vectoriales para lograr representaciones que maximicen la diferenciación entre los diferentes patrones existentes en los conjuntos de datos. Para ello, trabaja con funciones kernel, tales como laplaciano, polinomial, entre otros. Una de sus mayores ventajas es que trabaja óptimamente cuando el número de atributos es mayor al número de ejemplos. No obstante, su intepretación es compleja y presenta problemas a la hora de generalizar los comportamientos, lo cual se traduce en problemas de sobreajuste.
Métodos de ensamble	Los métodos de ensamble combinan diferentes algoritmos o variadas exploraciones de un mismo algoritmo en un único método, con el fin de optimizar el desempeño del modelo. Ejemplos clásicos de este tipo de algoritmos son los random forest, quienes se basan en la generación de múltiples árboles de decisión seleccionando iterativamente ejemplos y atributos de manera aleatoria. Este tipo de algoritmos resulta ser más costoso computacionalmente, No obstante, ha trabajado de manera exitosa en diferentes problemas de la biotecnología. Otro tipo de algoritmos similares a random forest son Adaboost, Gradient Tree Boost, entre otros.
Naive Bayes	Se basa en métodos probabilísticos para estimar la probabilidad de ejemplos asociados a sus categorías o etiquetas, los ejemplos más comunes de funciones se asocian a distribuciones normales. Sin embargo, existen variaciones donde se emplean distribuciones binomiales. Su interpretación es sencilla debido a que se basa en criterios de rango. No obstante, la parametrización de este tipo de algoritmos depende del conjunto de datos inicial, lo cual afecta a la generalización de los comportamientos.

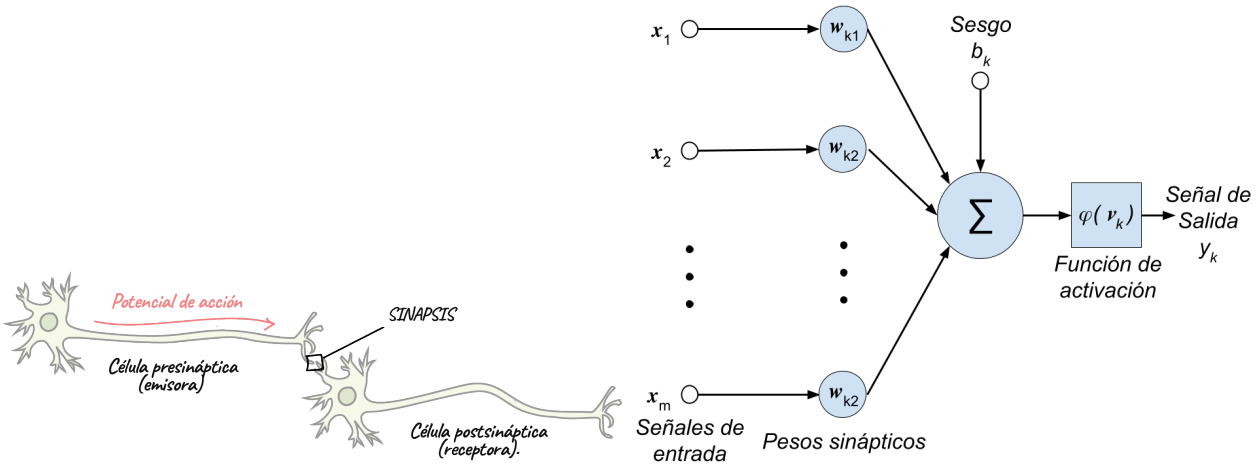
Tabla 1.2: Resumen general de los principales algoritmos de aprendizaje supervisado

1.4.3. Neuronas artificiales, redes neuronales y deep learning

Las neuronas artificiales son unidades de procesamiento de información fundamentales en una red neuronal. Son modelos matemáticos que se inspiran en las neuronas biológicas. En la Figura 1.5a y 1.5b se observa una comparativa entre lo que sería una neurona biológica y un esquema del modelo matemático de la misma.

Las neuronas artificiales cumplen con ciertas características:

- Realizan "sinapsis", proceso simbolizado por los enlaces que conectan entre unidades. Cada enlace tiene un valor de peso o fuerza. A diferencia de las neuronas biológicas,



(a) Neurona biológica (b) Neurona artificial

Figura 1.5: Comparación y visualización de Neuronas biológicas y artificiales

este peso puede tomar valores positivos o negativos.

- Poseen un agregador para sumar las señales de entrada, ponderado por los respectivos pesos de la neurona. Esta operación es conocida como combinación lineal.
- Tiene una función de activación. Esta limita la amplitud de la salida a un rango finito establecido. Es común que dicho rango sea dentro del intervalo [0, 1], o [-1,1], aunque hay excepciones.
- Puede poseer un parámetro externo llamado sesgo o *bias* especificado. Este puede tomar valores positivos o negativos, y se encarga de "desplazar" el modelo para mejorar la precisión.

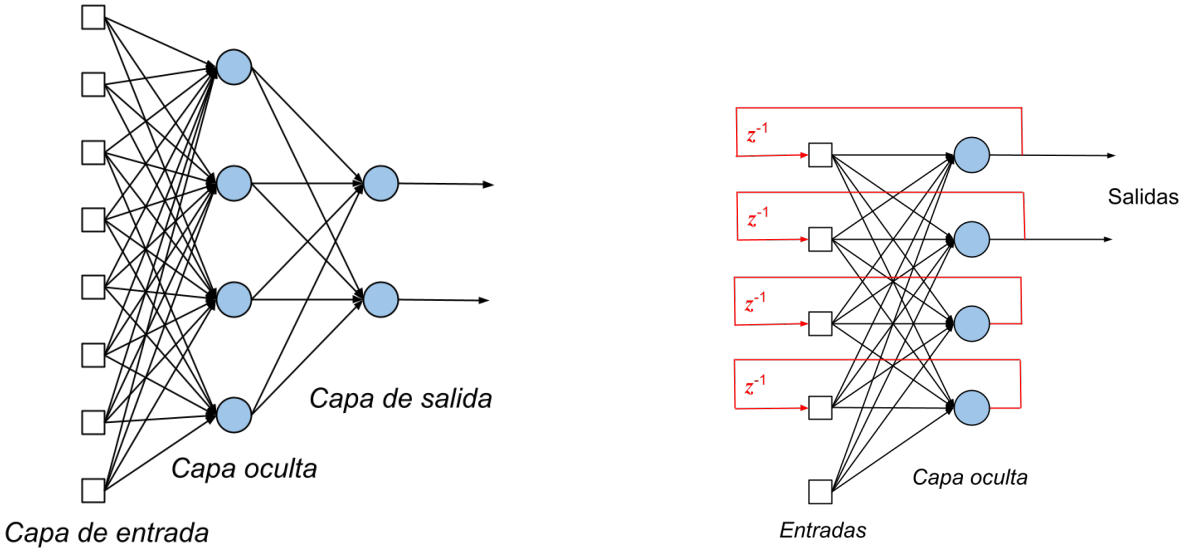
Las redes neuronales, como su nombre lo indica, son conjuntos interconectados de neuronas artificiales. Estas transmiten señales a través de sus relaciones, empleando funciones lineales (asociadas al peso de la interconexión w_k, también llamadas relaciones sinápticas) y no lineales (funciones de activación φ).

La primera red neuronal fue propuesta por Franck Rosenblatt en su libro *Perceptron* [142], cuyo modelo recibe el mismo nombre. Esta es una red de una sola neurona, similar a lo que se expone en la Figura 1.5b, utilizando la función escalón como función de activación. El perceptrón es un modelo de clasificación que utiliza hiperplanos como limitadores para cada una de las clases, similar a lo que implementa *support vector machine*. De esta forma, para nuevos datos, el modelo clasificará dependiendo de en qué lado del hiperplano se encuentre [70].

Se pueden generar múltiples arquitecturas o disposiciones de redes, principalmente asociadas al algoritmo de aprendizaje a utilizar. En general, existen dos tipos de arquitecturas de redes, las cuales se muestran en las Figuras 1.6a y 1.6b [70].

El ajuste de pesos en las redes neuronales se da mediante el algoritmo de *backpropagation*. Este es capaz de ajustar los parámetros del modelo utilizando el descenso del gradiente, el

cual corresponde a la minimización de la función de costo, utilizando derivadas parciales. La información en el algoritmo se dirige desde la última hacia las primeras capas, con el fin de analizar el error en la predicción y modificar aquellas neuronas que tengan mayor incidencia en él [143].



(a) Redes Feedforward

(b) Redes Recurrentes

Figura 1.6: Arquitecturas más comunes en las redes neuronales.

Finalmente, dentro de la enorme cantidad de algoritmos de *machine learning*, en el año 2006 surge un subconjunto de ellos bajo el nombre de *deep learning* o aprendizaje profundo. Estos algoritmos implementan nodos de procesamiento a través de capas jerárquicas llamadas *Redes neuronales artificiales*, con el objetivo de procesar la información de forma paralela y no lineal. Los algoritmos de *deep learning* pueden adaptarse a problemáticas de aprendizaje Supervisado y No Supervisado, lo que posibilita su aplicación en múltiples campos de investigación [167]. La profundidad viene definida por las diferentes formas en las que se pueden armar las arquitecturas. Además, de que las diferentes estrategias de generación de arquitecturas, permite resolver diferentes problemas. En la Tabla 1.3 se resumen de manera general los tipos de arquitectura más comunes y su utilidad.

Tipo de arquitectura	**Descripción general**	**Aplicaciones en ingeniería de proteínas**
RNN	Recurrent Neural Network	Desarrollo de sistemas autoencoders
CNN	Convolutional Neural Network, los ejemplos se representan como imágenes para poder identificar patrones en ellas	Predicción de funciones enzimáticas
LSTM	Long short time memories, empleada para manipular mensajes en memoria para un intervalo de tiempo	Modelos de clasificación para generación de secuencias
GNN	Graph Neural Networks, emplea estructuras de grafos como input para sistemas de predicción	Clasificación de plegamientos de proteínas.
G-CNN	Combina estructuras de grafos con procesos de convolución para poder desarrollar sistemas de predicción	Predicción de sitios activos en enzimas

Tabla 1.3: Arquitecturas clásicas de *deep learning* y sus aplicaciones en ingeniería de proteínas

1.4.4. Desempeño y evaluación de los algoritmos

Evaluar el desempeño de los modelos predictivos generados permite determinar cuán correcto fue el entrenamiento. Dependiendo del tipo de predicción (si se clasifican ejemplos o predicen nuevos valores) se emplean diferentes medidas de desempeño. A continuación, se resumen brevemente las medidas de desempeño más aplicadas.

Modelos de clasificación

Accuracy

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{1.1}$$

Precisión

$$\frac{TP}{TP + FP} \tag{1.2}$$

Recall

$$\frac{TP}{TP + FN} \tag{1.3}$$

F-score

$$\frac{2 \times Precision \times Recall}{Precision + Recall} \tag{1.4}$$

Medidas de regresión

Pearson's coefficient

$$\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\ \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1.5}$$

Kendall's τ rank

$$\frac{(number\ of\ concordant\ pairs) - (number\ of\ discordant\ pairs}{\frac{n(n-1)}{2}} \tag{1.6}$$

R score

$$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{1.7}$$

Spearman's rank

$$\frac{cov(rg_X), rg_Y}{\sigma_{rg_X}\sigma_{rg_Y}} \tag{1.8}$$

1.4.5. Principales problemas en el entrenamiento de modelos

De manera general, existen diferentes problemas a la hora de entrenar modelos predictivos, los cuales se resumen a continuación.

1. **Problemas de dimensionalidad**. Estos problemas están estrechamente relacionados con el conjunto de datos, es decir, a la cantidad de ejemplos y de atributos existentes en él. Si hay demasiados atributos y pocos ejemplos implica que se deben aplicar reducciones de dimensionalidad, mientras que si hay demasiados ejemplos y pocos atributos, se recomienda seleccionar ejemplos aleatorios desde el conjunto de datos para entrenar.
2. **Problemas con cantidad de ejemplos**. En reiteradas ocasiones, la cantidad de ejemplos en un conjunto de datos limita el entrenamiento de los modelos y la aplicación de técnicas de *machine learning*. Para prevenir este problema, es necesario evaluar la factibilidad del desarrollo de los modelos, con el fin particular de generalizar los comportamientos.
3. **Sobre ajuste**. Este problema es uno de los más complejos de trabajar, ya que evalúa cómo se comporta la generalización del modelo para predecir nuevos ejemplos. Para ello, técnicas como la validación cruzada y evaluaciones estadísticas, permiten determinar la existencia de sobre ajuste. Además, de la facilidad de diseñar tasas de evaluación del sobre ajuste, comparando las medidas de desempeño en las etapas de entrenamiento y la validación.

1.4.6. Generando un modelo predictivo

Una vez comprendidas las diferentes características, problemáticas y componentes del *machine learning*, es posible analizar un pipeline general para desarrollar modelos predictivos. La Figura 1.7 muestra las etapas clásicas en el desarrollo de modelos aplicando métodos de *machine learning*.

A partir de un conjunto de datos inicial es necesario en una primera etapa procesarlo, esto incluye quitar elementos nulos, eliminar repetidos y escalar a una misma distribución con el fin de disminuir posibles errores que causen atributos específicos.

En una segunda etapa se debe representar numéricamente los atributos categóricos para poder emplear los algoritmos de *machine learning*. Luego, con el conjunto de datos ya procesado se aplican los algoritmos de aprendizaje dividiendo el conjunto de datos en validación y entrenamiento. Esta división se hace para prevenir el sobre ajuste. Durante el entrenamiento, se aplican estrategias de validación cruzada con el fin de prevenir el sobre ajuste, en esta etapa se aplica un algoritmo con sus hiperparámetros de configuración.

Finalmente, se evalúan tanto el desempeño del modelo como el sobre ajuste, comparando el rendimiento en la etapa de entrenamiento como en la validación y cuantificando la diferencia entre ellos. De esta forma, resulta un modelo entrenado de manera clásica.

Si se desea entrenar un modelo aplicando estrategias de *deep learning*, es necesario en una

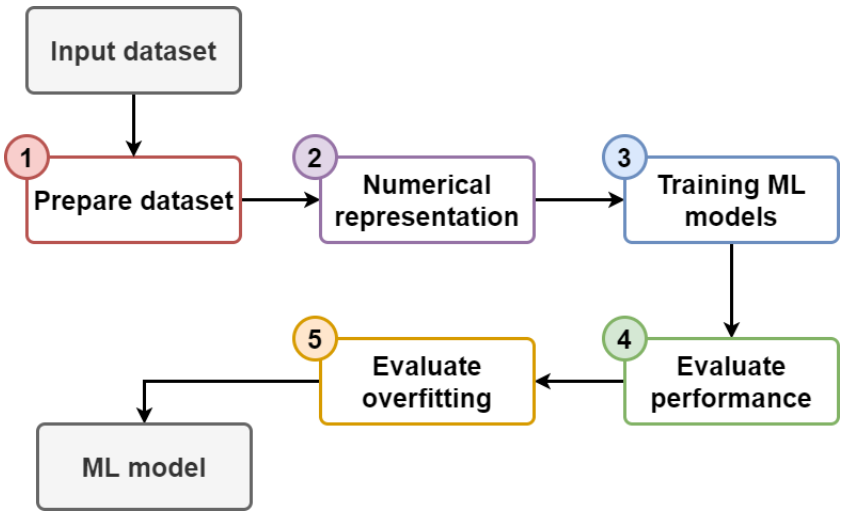


Figura 1.7: Etapas generales en el desarrollo de modelos predictivos aplicando estrategias de *machine learning* clásico.

primera instancia evaluar la representación del conjunto de datos y la cantidad de ejemplos existentes. Normalmente, si son representaciones vectoriales, se debe contar con una cantidad de ejemplos considerable con el fin de asegurar la generalización y la facilidad del aprendizaje dentro de la red. Por otro lado, si el conjunto de datos inicial corresponde a imágenes, es posible desarrollar estructuras matriciales para fomentar el entrenamiento, mientras que para el caso de redes interconectadas, normalmente se prefiere el uso de estructuras de grafos. Una vez identificada la forma en que se representará el conjunto de datos, es necesario seleccionar la arquitectura a emplear, la cual, no solo depende del tipo de información de entrada, sino que también del tipo de problema a resolver. Por ejemplo, si se desea trabajar con pronósticos y series de tiempo, normalmente se emplean las arquitecturas LSTM, mientras que para clasificar imágenes o datos matriciales se trabaja con CNN, en el caso de datos vectoriales se emplean redes RNN, lo cual es ampliamente utilizado en el caso de clasificación de documentos, finalmente, en el caso de grafos, se debe emplear arquitecturas basadas en GNN o GCNN. Una vez definida las arquitecturas, es necesario definir las capas que formarán la red y cómo fluirá la información. Luego, definir los hiperparámetros de configuración, los modelos optimización y las métricas de interés, así como también la evaluación de uso de capas de suavizado para la disminución de ruido. Tal como se observa, el problema es complejo y requiere una fuerte especialización en este contexto.

1.4.7. Algoritmos clásicos de aprendizaje no supervisado

Tal como se nombró previamente, el objetivo fundamental de los algoritmos de aprendizaje no supervisado consiste en la identificación de patrones o separación del conjunto de datos en grupos con características y propiedades definidas. La Figura 1.8 muestra una comparación visual de diferentes algoritmos empleados para el particionamiento de diferentes conjuntos de datos. La representación 2D facilita la visualización de los resultados y cómo los algoritmos se comportan. Normalmente, esta representación se hace mediante transformaciones espaciales asociadas a técnicas de reducción de dimensionalidad, tales como el uso de

Principal Component Analisis (PCA) y sus variaciones no lineales. En el ejemplo expuesto en la Figura 1.8, la última fila, representa un conjunto de datos nulos, es decir, ejemplos sin variación clara entre ellos, lo que, intuitivamente, permite suponer que el desarrollo de particiones no es posible. Sin embargo, algoritmos como *Affinity propagation*, o *MiniBatch K-means* si identifican grupos, esto es debido principalmente a la naturaleza de cómo funcionan y a su posible hiperparametrización.

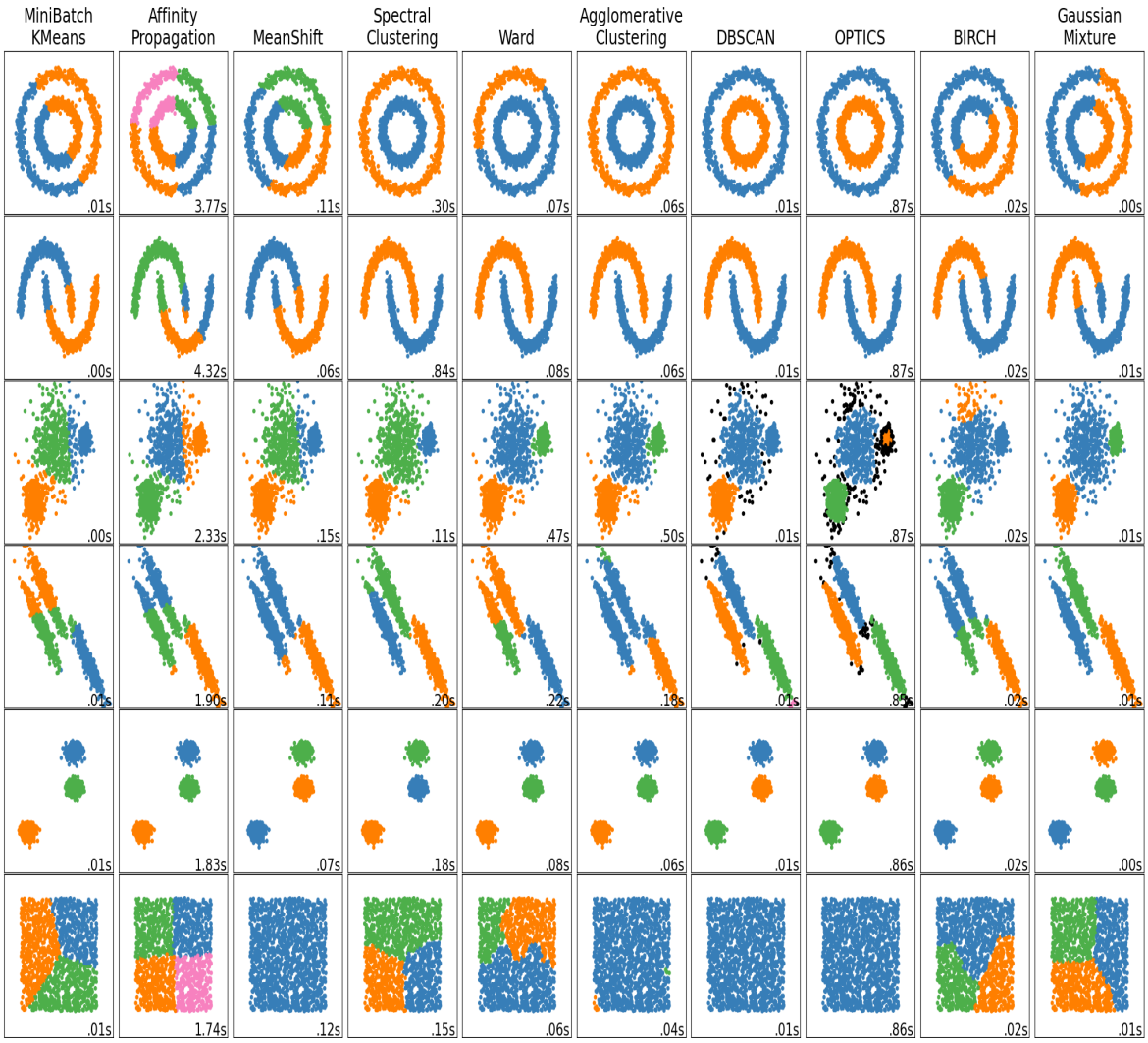


Figura 1.8: Representación 2D del funcionamiento de los algoritmos de aprendizaje no supervisado en diferentes datasets, extraída desde *scikit-learn* [125]

Algoritmos como *K-means*, *Affinity propagation* y jerárquicos (*Ward*) reciben dentro de sus hiperparámetros de configuración la cantidad de particiones deseables. El funcionamiento general entre ellos es el uso de métricas de distancia para poder comparar los ejemplos en un conjunto de datos. No obstante, conceptos como centroides y agrupación o división son empleados por los algoritmos jerárquicos. Sin embargo, las formas de agrupación y estrategias son similares, siempre contemplando el concepto de distancia entre ellos.

Por otro lado, algoritmos como *DBSCAN* y *Optics* no trabajan con hiperparametrización y tienen la ventaja que no requieren indicar el número de particiones. Estos algoritmos

también trabajan con estimación de distancia. Sin embargo, cada ejemplo, se toma como una esfera en el espacio, lo cual entrega espacialidad a la estimación de las distancias. A su vez, algoritmos como *Mean Shift* permiten emular el paso de mensajes, siendo eficientes a la hora de trabajar con conjuntos de datos dispersos [77].

Finalmente, existen algoritmos como Espectral y *Gaussian*, los cuales trabajan con transformaciones de datos y representaciones en forma de kernel matriciales para poder obtener particiones según parámetros de distribución. Por último, métodos basados en *deep learning* como los *Self Organization Maps* (SOM) también forman parte de métodos no supervisados, siendo una de sus peculiaridades el uso de transformaciones para disminuir el espacio dimensional y aplicar técnicas de división para poder obtener grupos en un espacio compacto o reducido.

1.4.8. Evaluación y desempeño

Tal como se logra apreciar, son diferentes las técnicas o estrategias que facilitan la identificación de patrones en conjuntos de interés, presentando distintas ventajas y desventajas dependiendo de los conjuntos de datos, representación numérica y elementos asociados al preprocesamiento inicial de los datos. Una de las formas más simples de determinar si las particiones son correctas es contemplando las métricas de desempeño asociadas a los algoritmos de clustering.

En particular, existen diferentes métricas como la homogeneidad, la completitud, *v-measures*, así como también los coeficientes de *mutual information* y *rand index*. No obstante, las desventajas que presentan estas métricas se asocian a que las particiones deben encontrarse definidas, es decir, los datos deben venir etiquetados. Pese a este requerimiento, este tipo de técnicas puede emplearse con el fin de evaluar si las particiones provenientes por el etiquetado de datos se encuentran separadas o será fácil la predicción o el entrenamiento para modelos predictivos (aprendizaje supervisado).

A diferencia de las métricas previamente nombradas, los coeficientes de siluetas y las métricas de *Calinkski-Harabasz index* son las medidas de desempeño más utilizadas y permiten evaluar la separación generada en cuanto a información, es decir, evaluar correlaciones entre los ejemplos de diferentes grupos para determinar si existen relaciones de dependencias. Los coeficientes de siluetas presentan valores entre 0 y 1, mientras que los *Calinski-Harabasz index* poseen valores entre un conjunto real. Con el fin de interpretar y comprender estos indicadores, a mayor valor, mejor es la separación generada.

La Figura 1.9 izquierda, muestra una representación de los coeficientes de siluetas a modo de ejemplo para la evaluación de un conjunto de datos con 4 particiones. A su vez, a la derecha en la Figura 1.9 se muestra la representación de los grupos. Una manera simple de ilustrar las separaciones entre los grupos es representar las dos primeras propiedades o características. Sin embargo, con el fin de hacer este proceso más eficiente, métodos como el Análisis de componentes principales se emplean para representar las propiedades en un espacio de maximización de información.

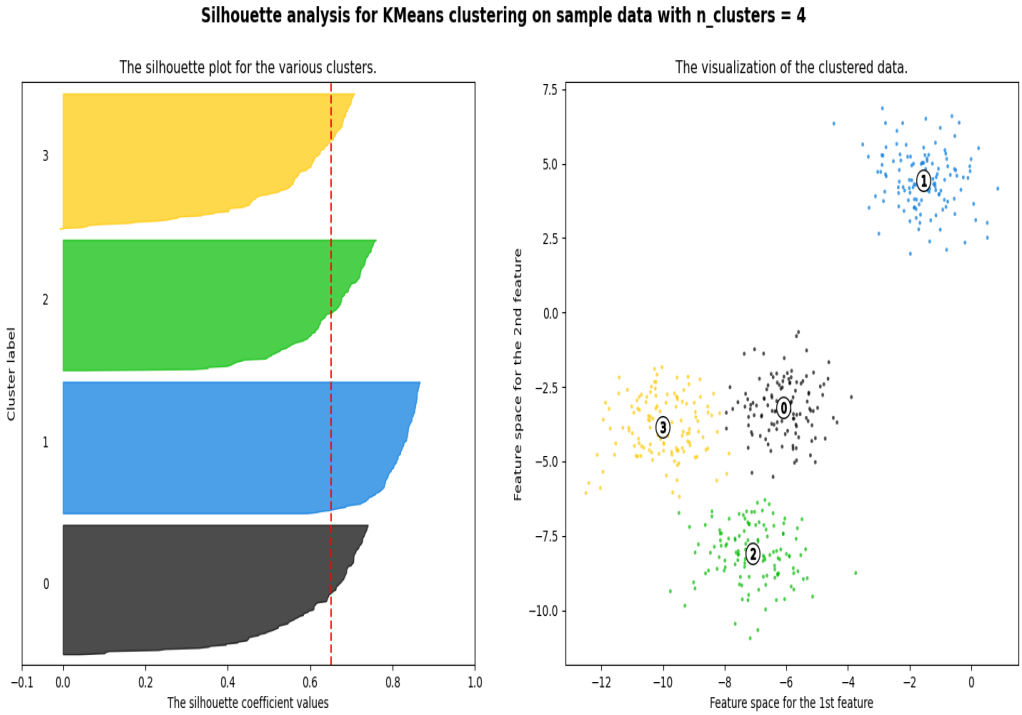


Figura 1.9: Esquema representativo de los resultados obtenidos por el Coeficiente de siluetas y la visualización de los grupos, extraída desde *scikit-learn* [125]

1.5. Aplicaciones del machine learning en ingeniería de proteínas

Las técnicas de *machine learning* y *data mining* han sido aplicadas en diversos estudios de ingeniería de proteínas. Variadas herramientas y estrategias se han desarrollado para resolver diferentes tareas específicas, jugando un rol fundamental en el avance científico de esta rama.

Tareas como reconstrucción de fitness [16, 97] y soporte a la evolución dirigida [177, 174] han sido las más frecuentes y en las que recientemente se han aplicado técnicas de *deep learning* para poder desarrollarlas. Sin embargo, se han empleado métodos de *machine learning* para tareas más específicas o variadas, contemplando no solo sistemas de clasificación, sino que también identificación de patrones y métodos de representación numérica de las secuencias amino acídicas [135, 107].

A continuación, se describe brevemente el estado de arte más reciente de las aplicaciones del *machine learning* en ingeniería de proteínas. Además, se discuten los principales desafíos y cómo se pueden abordar para generar un aporte a estos recientes avances.

1.5.1. Machine learning y deep learning como soporte de biología estrucrural

Sin lugar a duda, el mayor avance a la fecha en el ámbito de biología estructural ha sido el diseño e implementación de *AlphaFold* [6], el cual, combinando métodos de *deep learning* usando arquitecturas CNN con información de evolución filogenética y mapas de contacto, lograron predecir la estructura secundaria de las proteínas con una alta eficiencia y rendimiento. No obstante, una herramienta que cumple el mismo objetivo es *RoseTTAfold* [11], la cual sigue una estrategia similar a *AlphaFold*. Pero, empleando arquitecturas *N-D track-attention* como estrategia de aprendizaje.

Por otro lado, empleando *transfer learning* y arquitecturas LSTM, se han diseñado sistemas predictivos para evaluar el plegamiento de membranas [169], así como también la interacción de complejos proteína-proteína [52, 3]. No obstante, el desarrollo satisfactorio de métodos para los problemas de interacción es un problema latente y representa el siguiente escalón de desafíos en la biología estructural para los métodos de *machine learning*, siendo abordado actualmente por versiones recientes de *AlphaFold* [6] que permiten la predicción de multímetros.

Pese a los grandes avances que han presentado estas herramientas para la biología estructural, existen complicaciones en la utilización de estas, principalmente por el costo computacional que implica su uso, lo cual disminuye las posibilidades de su aplicación. Además, la generación de modelos estructurales proteína-ligando y proteína-proteína sigue siendo un problema y desafío latente, así como también el análisis de proteínas de membrana y la evaluación de interacción de grandes complejos proteicos con el fin de entender las dinámicas moleculares existentes en problemas clásicos con las reacciones enzimáticas.

1.5.2. Machine learning y deep learning como soporte para el diseño de nuevas secuencias

Diferentes métodos computacionales basados en *machine learning* o *deep learning* han sido implementadas Como herramientas de diseño, dentro de las principales, se encuentran las aplicaciones como soporte a técnicas de diseño de variantes, tales como [177, 174] quienes proponen el entrenamiento de modelos para guiar la selección de variantes en el marco de la evolución dirigida. Por otro lado, existen herramientas como *EcNet* [97] y *Unirep* [4] que se han centrado en definir estrategias para representar las secuencias con el fin de mejorar el diseño de proteínas y contribuir específicamente a la reconstrucción de fitness, es decir, facilitar la identificación de mutaciones o variantes que aumenten la propiedad de interés para una proteína en específico. Por otro lado, la aparición de las estrategias de *deep generative models*, ha facilitado la generación de secuencias para mapear conocimiento a priori experimentalmente [176, 119]. No obstante, requiere de un número elevado de secuencias input para poder ser aplicadas de manera correcta.

Pese a los enormes avances tanto en construcción de fitness como en generación de secuencias, el diseño de mutaciones o variantes con propiedades deseables es un problema aún

latente, siendo uno de los grandes desafíos de la ingeniería de proteínas de los últimos 40 años y el cual será uno de los focos de interés de este trabajo de tesis doctoral.

1.5.3. Aplicaciones del *deep learning* y el *machine learning* para resolver tareas específicas de ingeniería de proteínas

Dentro de las principales aplicaciones del aprendizaje supervisado se encuentran las técnicas de clasificación de funciones de proteínas, como la evaluación de proteínas de unión a DNA o RNA[129], para los cuales normalmente se han empleado métodos basados en *random forest* y recientemente la aplicación de *deep learning*, en particular, enfoques basados en *convolutional neural network* [129]. Otros enfoques han permitido la predicción de estabilidad de mutaciones empleando algoritmos de *support vector machine* [24] y la estimación de diferentes propiedades como enantioselectividad y accesibilidad al solvente, empleando métodos como *partial least square* [22] y métodos de ensamble [182]. Recientes avances en *deep learning* han facilitado la aplicación de estas estrategias en el *drug discovery* [187], la clasificación de antibióticos terapéuticos [103, 63], la anotación y clasificación de nuevas secuencias [15], así como el desarrollo de *frameworks* para tareas genéricas [179, 155].

La lista de tareas y aplicaciones posibles a resolver es extensa y variada, lo cual denota que a la fecha no se han logrado los objetivos de diseñar métodos genéricos de clasificación, así como también *frameworks* de entrenamiento de modelos, transferencia de aprendizaje y evaluación del límite del poder predictivo de los modelos, lo cual genera un punto de desarrollo importante y de interés para la comunidad científica.

1.6. Los mayores desafíos de la ingeniería de proteínas en las últimas décadas

Los problemas y desafíos en la ingeniería de proteínas son variados. Sin embargo, el mayor desafío de los últimos cuarenta años se ha centrado en el diseño de proteínas con propiedades deseables. A pesar de la existencia de protocolos definidos como la evolución dirigida y el diseño racional, y sus mejoras incorporando métodos basados en aprendizaje de máquinas, el problema aún persiste.

Más en detalle, los principales problemas que serán abordados en esta tesis de doctorado se resumen a continuación.

1.6.1. Representaciones numéricas, cuál es la mejor alternativa?

A menudo, el uso de secuencias lineales de proteínas se relaciona a la identificación de patrones o evaluación de variantes para una misma proteína. Actuales herramientas bioinformáticas permiten el uso de la secuencia de manera directa y por medio de alineamientos de

secuencias o modelamiento a través del uso de Cadenas de Márkov, facilitan el reconocimiento de patrones o la evaluación de mutaciones. No obstante, para la aplicación de métodos basados en inteligencia artificial, ya sea la identificación de clúster o el entrenamiento de modelos predictivos, se requiere generar representaciones numéricas de la secuencia.

Existen diferentes formas de representar numéricamente las secuencias, tales como *one hot encoding*, propiedades fisicoquímicas [107], así como estrategias más recientes basadas en *text mining* [182] y la generación de modelos de aprendizaje de representaciones [135]. A pesar de ello, no existe un consenso asociado a qué técnica utilizar. Cada una presenta sus pros y contra. No obstante, la cantidad de información involucrada varía entre ellas, así como también la forma de procesamiento, tiempo de cómputo, entre otras. Sin embargo, a mayor información, incrementa el número de dimensiones a tratar, aumentando la complejidad del problema. Esto implica, utilizar técnicas de reducción de dimensionalidad para seleccionar las dimensiones con mayor variabilidad en el conjunto de datos. Además, recientes avances han combinado las representaciones de propiedades fisicoquímicas con transformaciones al espacio de frecuencias con el fin de adquirir una mayor visibilidad de las interacciones y los patrones funcionales [107, 157]. Sin embargo, problemas como la selección de propiedades, la definición de estrategias de transformación y cómo reconstruir el espacio latente, forman parte importante de las interrogantes a resolver.

En vista de las necesidades de desarrollo de modelos de clasificación/regresión y la generación de sistemas de clustering para secuencias lineales de proteína, con el fin de apoyar al diseño de proteínas, análisis de variantes e inclusive caracterización de secuencias, sin tener conocimiento sobre su estructura. Se propone el uso de transformadas de Fourier como método de digitalización de propiedades fisicoquímicas para el desarrollo de conjuntos de datos que permitan ser entrenados para el desarrollo de estimadores o identificar patrones, siendo el tema central a abordar en el capítulo 2 y sus aplicaciones para entrenamiento de modelos ensamblados en el capítulo 3, como soporte de estrategias de análisis de espacios latentes en el capítulo 4, así como también su aplicación en identificación de patrones asociados al trabajo a futuro que se está desarrollando.

1.6.2. Estudio de mutaciones puntuales, cómo podemos caracterizar las mutaciones?

El desarrollo de modelos de clasificación y/o regresión, es uno de los temas más recurrentes en el campo de la minería de datos y el aprendizaje de máquinas. Sin embargo, el hecho de asociar mutaciones a una respuesta, conlleva al problema de cómo caracterizarla, con el fin de alimentar a los algoritmos para ser entrenados.

A raíz de esto, cuáles son los mejores descriptores para una mutación?, desde qué puntos de vista se puede hacer una caracterización? Y cuáles son más relevantes?, son interrogantes que se presentan a la hora de abordar su representación, siendo problemas que han sido tratados desde un largo tiempo sin lograr generar un consenso o una forma general de diseñar tal caracterización.

En un gran número de trabajos, en los cuales se ha evaluado la estabilidad de proteínas en

torno a la mutación, se han utilizado descriptores termodinámicos y de ambiente para poder representar el cambio amino acídico [60, 118]. A pesar de que los desempeños de los estimadores han sido aceptables y relativamente altos. Esta caracterización ¿podrá ser utilizada para mutaciones asociadas a riesgo clínico?, ¿Existirá una correlación entre la respuesta y las variables de interés?, ¿Cómo afecta al desempeño del modelo la existencia de diferentes ejemplos asociados a distintas proteínas en un único conjunto de datos?, etc., son interrogantes que nacen a la hora de plantearse la situación.

Dado a lo anterior, y con el objetivo de generar un aporte significativo al desarrollo de estimadores basados en aprendizaje de máquinas, se ha propuesto adicionar el concepto de filogenia a la descripción de mutaciones y disgregar los conjuntos de elementos para ser tratados por proteínas independientes, esto con el fin de generar modelos de clasificación y/o regresión proteína-específicos, los cuales puedan ser aplicados a diferentes respuestas de interés ya sea: efectos en mutaciones, estabilidad, actividad, productividad, etc., siendo este, el tema central a abordar en el capítulo 5.

1.6.3. Diseñar mutaciones, proponiendo nuevas estrategias de diseño

Diseñar proteínas de manera eficiente, con una identificación adecuada de la propiedad en estudio o funcionalidad a adicionar, sin incurrir en grandes costos económicos y de recursos, es uno de los desafíos más relevantes de la ingeniería de proteínas. Como se nombró previamente, son dos los enfoques que se utilizan actualmente: Evolución dirigida y diseño racional.

Ambas técnicas tienen sus ventajas y desventajas. No obstante, poseen en común una demanda en tiempo elevada y se requiere de conocimientos profundos sobre la estructura para poder diseñar las mutaciones, al menos, para el caso de diseño racional.

Enfoques computacionales han sido propuestos, con el fin de minimizar los costos económicos, contemplando evaluaciones energéticas asociadas a los residuos y cómo estos afectan a la estabilidad. No obstante, no pueden ser utilizados en secuencias lineales. Además, dejan de lado el concepto filogenético en el estudio, resultando un gap entre ambos puntos de vista. Por otro lado, métodos basados en la minería de datos, solo se han centrado en identificación de residuos o el entrenamiento de modelos para predecir estabilidad o propiedades simples de predicción.

Recientemente, el *machine learning directed evolution* [174] y los métodos semi-racionales de diseño aprovechan las ventajas del *deep learning* o *machine learning* para generar simulaciones basándose en modelos predictivos y así, apoyar o dirigir las técnicas de diseño. Sin embargo, presentan los mismos problemas asociados a la representación numérica, caracterización de los conjuntos de datos y cómo identificar puntos claves o de relevancia, así como secuencias que maximicen la propiedad de interés, sin tener que incurrir en métodos de heurística y costosos sistemas computacionales.

A partir de lo anterior, y con el fin de generar un aporte significativo en el área de diseño, se ha considerado esta problemática como un foco central y culminante para el desarrollo de este trabajo, proponiendo así, la implementación de métodos computacionales, basados en

técnicas de minería de datos y aprendizaje de máquinas, que permitan proponer mutaciones o variantes dado un conjunto de secuencias con respuesta conocida. Trabajando con variados métodos de representación numérica y emulando variantes en el espacio latente basándose en el reconocimiento de sitios de interés en la proteína objetivo. Toda esta problemática, el planteamiento de la metodología y qué se utilizará para llevar a cabo, se abordará en el capítulo 4.

1.7. Hipótesis

Con base en las herramientas computacionales existentes y a los problemas expuestos previamente en este capítulo. Además, tomando en consideración los avances en minería de datos y aprendizaje de máquinas a la fecha. Se propone la siguiente hipótesis.

Combinar representaciones numéricas de secuencias de aminoácidos optimizadas para la aplicación de algoritmos de machine learning, junto con sistemas predictivos basados en aprendizaje por ensamble, mejora los rendimientos de los modelos predictivos en tareas de ingeniería de proteínas, facilitando su uso para la elaboración de estrategias de diseño de proteínas con propiedades deseables y estudio de mutaciones.

1.8. Objetivos

1.8.1. Objetivo general

Elaborar estrategias computacionales para el diseño de proteínas con propiedades deseables y estudio de mutaciones con aplicaciones en ingeniería de proteínas, soportadas por optimización de representaciones numéricas y sistemas predictivos entrenados mediante técnicas de aprendizaje por ensamble.

1.8.2. Objetivos específicos

1. Crear estrategias computacionales para representar numéricamente las secuencias de proteínas a partir de propiedades fisicoquímicas semánticamente seleccionadas, con el fin de facilitar la aplicación de estrategias de inteligencia artificial.
2. Construir métodos computacionales para el entrenamiento de modelos predictivos empleando aprendizaje por ensamble combinado con representaciones numéricas basadas en propiedades fisicoquímicas seleccionadas por sentido semántico.
3. Elaborar protocolos de diseño de secuencias con propiedades deseables a partir del estudio de espacios latentes estadísticos provenientes de la representación numérica de las secuencias, como soporte para las herramientas de diseño experimentales.

4. Diseñar métodos computacionales para el estudio de mutaciones puntuales y la identificación de sitios relevantes para mutagénesis sitio-dirigida, combinando los puntos de vista filogenéticos, termodinámico y estructural.

Tanto el objetivo general como los tres objetivos específicos propuestos serán abordados en los capítulos dos al cinco, donde se expondrán las diferentes metodologías desarrolladas y las aplicaciones de estas en diferentes problemas comunes de la ingeniería de proteínas.

Capítulo 2

Representaciones numéricas

Uno de los principales problemas a la hora de aplicar técnicas de machine learning y data mining en ingeniería de proteínas, es la representación numérica de las secuencias de aminoácidos.

Diferentes estrategias han sido desarrolladas para resolver este problema. Los primeros enfoques se basaron en la binarización de las secuencias empleando técnicas como *one hot* u *ordinal encoder* [21]. No obstante, la alta dimensionalidad del conjunto de datos y la falta de significado biológico son problemas recurrentes en estas estrategias de codificación. Alternativamente, el uso de las propiedades fisicoquímicas y termodinámicas de los aminoácidos ha sido ampliamente explotada como método de codificación de secuencias. Sin embargo, problemas como la selección de las propiedades y la falta de representatividad de la interacción entre los residuos han provocado falencia de información para los sistemas predictivos.

Nuevos enfoques han empleado estrategias de *text mining* basadas en *doc2vec* o *word2vec* para trabajar las secuencias como documentos y desarrollar sistemas de aprendizaje de representación numérica [182, 181]. No obstante, al igual que en los métodos previamente mencionados, carece de representación biológica o es demasiado compleja su interpretación. Otro enfoque interesante se ha basado en métodos soportados por teoría de grafos, siendo empleado con éxito en los últimos años [61]. Sin embargo, su uso es limitado, ya que se necesita la estructura de la proteína para aplicar estas representaciones y, por otro lado, no existe un consenso claro sobre la mejor estrategia de generación de los grafos. Por otro lado, a partir de las representaciones numéricas se han desarrollado técnicas para generar imágenes con el fin de incrementar las capacidades de identificación de patrones a partir de estrategias de convolución y de computación visual [145, 44]. Sin embargo, estas técnicas dependen principalmente de la representación numérica, lo cual denota los mismos problemas expuestos previamente asociados a dichas representaciones. Además, se necesita un gran volumen de datos para balancear el número de parámetros asociados al aprendizaje en las CNN.

Como alternativa, las estrategias basadas en codificaciones por propiedades fisicoquímicas han sido utilizadas en combinación con técnicas de procesamiento digital de señales para representar secuencias en el espacio de frecuencias [22, 157, 107]. Esta estrategia tiene la peculiaridad de homologar los comportamientos de una proteína a nivel estructural. Pese a esta ventaja, seleccionar qué o cuántas propiedades utilizar y de qué forma combinarlas es

un problema complejo, que a la fecha no ha sido resuelto. Debido a que las transformadas de Fourier facilitan la transformación de espacios reales a frecuencias, es recurrente su uso en procesamiento de señales, esto, sumado a las ventajas que integra este tipo de representaciones a la hora de simular los efectos de interacción entre los residuos [40], lo hacen parecer una estrategia interesante, de gran usabilidad y un método potente para representar numéricamente las proteínas [107, 156]. Sin embargo, la necesidad de identificar las propiedades claves para su transformación dificultan su usabilidad.

A pesar de la existencia de múltiples estrategias de representación numérica, el problema aún persiste, demostrando la necesidad de diseñar metodologías de codificación numérica de fácil comprensión y que faciliten la simulación de los comportamientos estructurales de la proteína para ser incluidos como información en posteriores desarrollos de sistemas predictivos.

En este capítulo, se abordarán las diferentes estrategias de codificación, nombrando sus ventajas y desventajas. Además, se propone una metodología para generar codificadores de propiedades fisicoquímicas, los cuales combinados con técnicas de procesamiento digital de señales, facilitan la representación numérica de secuencias con sentido biológico y permiten emular los posibles comportamientos estructurales con respecto a interacción. El método propuesto fue evaluado midiendo el desempeño de modelos de predictivos y comparando los resultados con estrategias clásicas, representación como objetos y técnicas previamente reportadas, logrando presentar mejores resultados en diferentes tareas predictivas de ingeniería de proteínas, demostrando la usabilidad y las ventajas de la técnica propuesta en esta metodología. Además, las representaciones fueron utilizadas como métodos de clustering para la identificación de patrones y como input para entrenamiento de modelos ensamblados y diseño de espacios latentes probabilísticos, lo cual se verá en detalle en los siguientes capítulos.

2.1. Estrategias y metodologías de codificación

Las estrategias de codificación buscan representar un vector categórico de manera numérica. Existen diferentes tipos de metodologías previamente desarrolladas. La Figura 2.1 genera un resumen general de las principales metodologías de representación numérica, las cuales pueden dividirse en generación de codificadores (Figuras A-D), representaciones de imágenes (Figura E) y representaciones de grafos (Figura F). Se alcara que no se nombran las representaciones con base en *natural language processing* y procesos de aprendizaje en la Figura 2.1 debido a que se expondrán más en detalle en las siguientes secciones del escrito. A continuación, se describen brevemente cada una de estas técnicas, así como también su uso en ingeniería de proteínas, sus ventajas y desventajas a la hora de interpretar los resultados y su representación.

2.1.1. One hot y ordinal encoder

One hot encoder, es una de las técnicas más utilizadas a la hora de codificar variables categóricas y se basa principalmente en la adición de columnas con respecto a las categorías

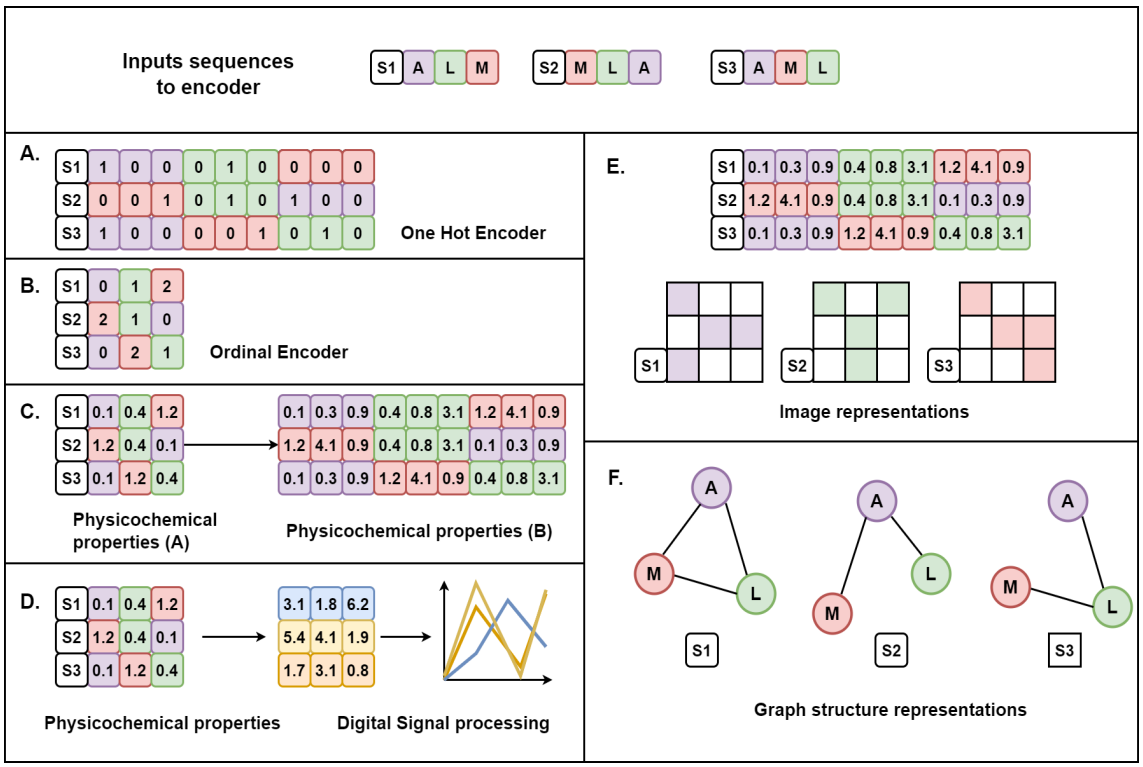


Figura 2.1: Esquema representativo de las principales estrategias de codificación y representación numérica aplicadas en ingeniería de proteínas.

existentes en un conjunto de datos [21].

Dado el vector x de tamaño n con m categorías, por definición, *One hot encoder* agrega al conjunto de datos $m-1$ columnas. Las nuevas columnas se completan con una binarización de los elementos, indicando si el elemento x_i posee la categoría m_j con un valor 1 y en caso contrario 0. En el caso de secuencias de proteínas, las categorías se consideran como los 20 residuos canónicos, de esta manera, los vectores codificados quedan de tamaño $n \times 20$. Esto puede ser resumido en la Figura 2.1 A.

Ordinal encoder, es una simplificación de *One hot encoder*, ya que, simplemente codifica las categorías con números en el conjunto $[0, m-1]$. Es decir, sea el vector x de tamaño n con m categorías y sea M el espacio de las posibles categorías con $M = [m_1, \cdots, m_m]$, y cuya codificación implica el vector $M' = [0, \cdots, m-1]$. $\forall$ elemento que $\in$ a x se obtiene su codificación a partir del elemento $M'(M(m_i))$ que corresponde a la codificación de la categoría en el espacio M [125]. La Figura 2.1 B, representa esquemáticamente este tipo de codificación.

Este tipo de codificaciones ha permitido implementar modelos predictivos para predecir *hot spot* en proteínas [124], así como también ha facilitado la implementación de metodologías para el diseño sintético de anticuerpos [173]. No obstante, estos modelos son relativamente antiguos, presentan varias falencias a la hora de su replicabilidad. Además, el uso de estas técnicas se relaciona a problemas asociados con la alta dimensionalidad de los vectores numéricos y la interpretación biológica. Por otro lado, en contextos como regularización de genes, ha permitido implementar redes de expresión considerando la binarización como una

estrategia eficiente de representación [96]. Sin embargo, no brinda información suficiente a modelos predictivos para asegurar el aprendizaje, ya que los patrones no son claramente definidos. Por otro lado, cuando las secuencias a codificar no presentan el mismo tamaño, es necesario aplicar técnicas como *zero-padding*[1] con el fin de ajustar todos los vectores a un mismo largo, lo cual es requisito a la hora de aplicar algoritmos de *machine learning*.

2.1.2. Frecuencias de residuos

Esta estrategia de representación numérica se basa en obtener la frecuencia de los residuos en la secuencia y reemplazarlos por dichos valores. Existen diferentes formas de implementar esta estrategia. Sin embargo, son dos las principales [120].

1. Para una secuencia S con r residuos se estima la frecuencia f de cada residuo r como $\frac{n_r}{L}$ donde n_r es el número de veces que aparece el residuo r en la secuencia S y L es el largo de la secuencia. Cada residuo r es reemplazado por su correspondiente valor f.
2. Dado el espacio E de los posibles residuos en una secuencia de proteína P, se genera un vector numérico de tamaño L donde L es el largo del espacio E. Luego, se estima la frecuencia f para cada residuo r en el espacio y se reemplaza el elemento E_i del espacio E por la frecuencia f calculada. En el caso de no existir el residuo r en la secuencia P, simplemente se completa con un valor 0.

En esta estrategia, cada residuo r es representado en un rango entre $[0, 1]$. Sin embargo, solo la información del conteo de residuos en la secuencia es insuficiente y no permite identificar patrones complejos en las proteínas asociados a las relaciones entre los residuos. Fallando específicamente en la representación de conjuntos de datos con mutaciones puntuales. Además, existen variaciones a este tipo de representación en las cuales, en vez de considerar solo un residuo r, se consideran dipéptidos o tripéptidos, lo cual busca identificar patrones filogenéticos en las secuencias a modo de brindar mayor información para su usabilidad. No obstante, dichas estrategias no han sido aplicadas a modelos predictivos, solo se han considerado para análisis filogenéticos y estadísticos [29, 160].

2.1.3. Uso de propiedades fisicoquímicas

Diferentes métodos experimentales han permitido obtener propiedades fisicoquímicas de los residuos para su caracterización. Bases de datos como *AAIndex* [82] han recopilado dichas propiedades y las han habilitado para su uso en variadas aplicaciones de ingeniería de proteínas [107, 157].

Este tipo de representación se basa en codificar un residuo r_i de una secuencia S empleando una o un conjunto de propiedades P (Ver Figura 2.1 C), de esta forma, el residuo r_i se representa por un vector $v = [P_1(r_i) \cdots P_n(r_j)]$, donde el tamaño del vector v es j depende del número de propiedades que se deseen aplicar.

[1] Completar con ceros hasta que todos los vectores tengan el mismo largo.

Este enfoque ha sido ampliamente empleado en la generación de descriptores para conjuntos de datos en ingeniería de proteínas [23, 25], así como también para elaborar sistemas predictivos de estabilidad de proteínas [8, 20, 128] y evaluación de enantioselectividad en mutaciones puntuales [22]. Así como también, en clasificación de antígenos para células tumorales [72].

Uno de los principales problemas a la hora de emplear esta estrategia recae en la selección de las propiedades a emplear, así como su cantidad, la elección de las propiedades más representativas y cómo combinarlas en caso de ser necesario. Si bien técnicas basadas en reducción de dimensionalidad e identificación de patrones han tratado de resolver este problema [58, 156], los patrones identificados no son consistentes en cuánto a la definición de propiedades, ya que, numéricamente, pueden tener una gran similitud. Pero, biológicamente, significados muy diferentes. Lo cual hace a este problema, complejo de resolver y un tema abierto de desarrollo a la fecha.

2.1.4. Natural language processing

Natural language processing (NLP) es una técnica que permite entender el contexto, la cohesión y la sintaxis dentro de un texto [34]. Estudios recientes han comenzado a aplicar métodos basados en NLP para la representación numérica de las secuencias de proteínas, con el fin de poder aplicarlas en sistemas predictivos o identificación de patrones complejos a partir de las secuencias de proteínas [182].

La base de este desarrollo se ha centrado en la generación de *autoencoders*, los cuales se definen como sistemas de aprendizaje que facilitan la codificación de secuencias en vectores numéricos continuos pertenecientes al espacio real y su decodificación [13]. La Figura 2.2 muestra una arquitectura clásica de los *autoencoders*, donde se aprecia claramente los componentes principales de un *autoencoder*: el codificador y el decodificador.

La forma en la que se entrenan estos sistemas predictivos es basándose en métodos de *text mining*, ya sea *doc2vec* o *word2vec* [85] y la subdivisión de la secuencia en segmentos denominados $k-$mers, los cuales constituyen sub secuencias de tamaño k. Estos $k-$mers son ingresados al sistema en forma de tripletes, donde el elemento central consiste en la secuencia a aprender a representar numéricamente, mientras que los segmentos laterales representan el contexto del mismo. La representación numérica de estos $k-$mers se centra en una binarización mediante *one hot*. Finalmente, el proceso de aprendizaje se centra en la representación numérica de las secuencias en un espacio real mediante el uso de *embeddings*, generando vectores comprimidos de información, donde su tamaño es parametrizable. No obstante, debido a que la idea principal es generar vectores únicos, normalmente se emplean tamaños que van desde los 760 hasta los 1900, como son los casos de los modelos pre-entrenados *Bert* y *Bubbler* [135]. De esta forma, estas técnicas permiten obtener vectores numéricos a partir de las secuencias lineales de los aminoácidos.

Diferentes aplicaciones en ingeniería de proteínas han sido desarrolladas bajo este foco del *text mining*, los principales desarrollos se han centrado en el uso de *embeddings* para entrenar modelos en ingeniería de proteínas [182], generar *autoencoders* para familias de secuencias de proteínas [135], y en otro contexto, aplicar estas metodologías para representar *SMILES*

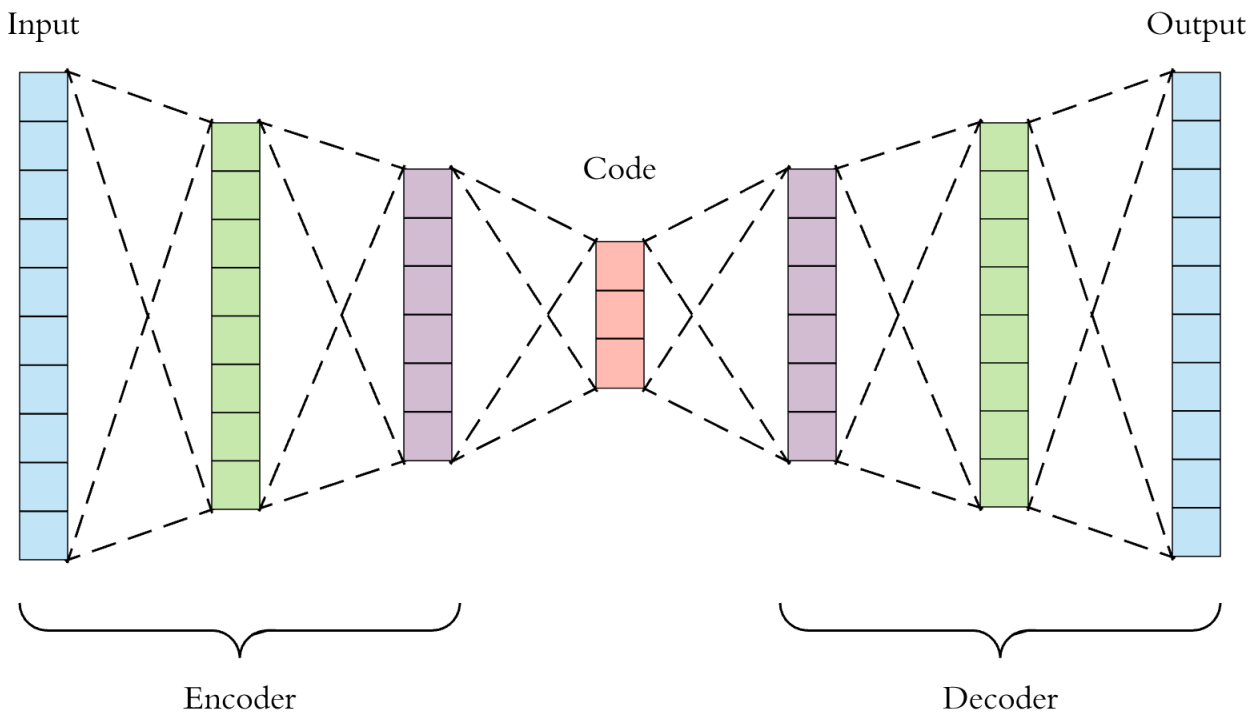


Figura 2.2: Esquema representativo de un *autoencoder* como método de desarrollo de vectores numéricos

[87].

Pese a los diferentes beneficios que otorgan este tipo de codificaciones, existen variadas problemáticas al momento de emplearlas.

1. Se requiere un número elevado de secuencias para asegurar la generalización del aprendizaje del *autoencoder*.
2. El costo computacional para su desarrollo es elevado, tanto para entrenar nuevos modelos como a la hora de emplear modelos pre-entrenados o herramientas como *TAPE* [135].
3. La interpretación biológica de los vectores es compleja de desarrollar. Debido al nivel de abstracción, normalmente no se suelen apreciar relaciones simples.
4. Pese a la existencia de herramientas como *TAPE* [135], el problema de la dimensionalidad aún persiste, dado que sus codificadores crean vectores de tamaños 760 y 1900, dependiendo del modelo empleado, lo cual, a la hora de clasificar secuencias de péptidos con tamaños de 5 a 150 residuos, denota un incremento importante de la dimensionalidad.

2.2. Imágenes y estructuras de grafos como estrategias de representación de proteínas

Más allá de la representación de las secuencias como vectores numéricos, se han implementado estrategias que facilitan su representación como objetos. Dentro de estas estrategias, se destacan el uso de estructuras de grafos y la aplicación de técnicas de procesamiento de imágenes. A continuación, se hablará brevemente de cada una de estas técnicas, su forma de trabajo, definiciones, características y principales problemáticas.

2.2.1. Estructuras de grafos aplicadas a representación de proteínas

Un grafo es un tipo de estructura que permite almacenar información no estructurada [150], se ha aplicado en diferentes estudios en ingeniería de proteínas, ya sea con el fin de identificar patrones, desarrollar modelos predictivos, o generar aplicaciones para el diseño de nuevas variantes [75, 76]. Sin embargo, su mayor explotación ha sido en estudios de *drug discovery* [61].

Matemáticamente, un grafo se puede definir como

$$G = (N, E) \tag{2.1}$$

Donde N representa el conjunto de nodos y E el conjunto de aristas, es decir, las conexiones existentes entre los nodos, las cuales pueden ser con sentido (grafos dirigidos) o sin sentido (grafos no dirigidos). Para representar proteínas, es necesario utilizar la estructura tridimensional de ellas, con el fin de que cada nodo N sea un residuo y las aristas representen las interacciones entre los residuos. La generación de nodos puede desarrollarse desde diferentes puntos de vista, dentro de los principales destacan.

1. Átomos como representación de nodos, en este sentido, se emplean los átomos de la estructura para generar un nodo.
2. Carbono α como representación de un residuo. Se emplea este átomo para representar a un residuo, contemplando que es átomo central de la estructura básica del aminoácido.
3. Centroides. Se define un átomo sintético, el cual corresponde al promedio de las coordenadas de las posiciones de cada átomo en el residuo, con el fin de emular un punto central en el residuo a representar.

Así como existen diferentes estrategias de construcción de nodos, existen variados métodos de generación de aristas, dentro de las cuales destacan.

1. Aristas generadas por estimación de distancia. Se estima una distancia euclidiana o variantes entre diferentes nodos. En un grafo *full connected*, la distancia puede ser

asociada como peso en las aristas. No obstante, normalmente es necesario emplear rangos de sentido biológico con el fin de conectar los nodos.

2. Estimación de interacciones electrostáticas débiles como evaluación de interconexión entre nodos. Esto es asociado principalmente a puentes de hidrógenos.

Dado a esto, es posible generar estructuras de grafos para resolver problemas de ingeniería de proteínas. No obstante, su desarrollo y aplicación puede presentar diferentes problemas. Primero, existe la necesidad de contar con la estructura tridimensional de la proteína. Por otro lado, a la hora de entrenar modelos predictivos, los problemas con la generalización y el sobre ajuste de los modelos resultan bastante complejos. Además, si el conjunto de datos se encuentra desbalanceado, normalmente los modelos generados tienden a un sobre ajuste, no siendo capaz de identificar diferencias sutiles que alteren la clasificación de una respuesta, por ejemplo, mutaciones puntuales no relacionadas con sitios activos o alostéricos en enzimas. Finalmente, la elección de las estrategias de generación de nodos y creación de aristas aún no está generalizada y es un tema importante de desarrollo pendiente en la actualidad.

2.2.2. Aplicaciones de imágenes a representaciones de proteínas

Una imagen es una representación matricial de píxeles combinados con información de espectros de colores RGB. No obstante, si solo se toma el componente de los píxeles, es posible definir patrones matriciales que representan imágenes, similar a las metodologías de *fingerprinters* [78, 102]. Teniendo esto en mente, es posible extrapolar estas estrategias combinadas con las representaciones numéricas para generar representaciones matriciales correspondientes a imágenes.

De esta forma, sea una secuencia S representada numéricamente con alguna estrategia de codificación, ya sea de las antes mencionadas o alguna nueva implementada, tal que se forma el vector R de dimensiones $(1, n)$. Dicho vector, mediante transformaciones espaciales, es posible redimensionarlo para generar una representación matricial M con dimensiones (m, n), la cual, puede ser representada como imágenes. Aplicando esta transformación espacial, las dimensiones de la matriz M deben ser las mismas para cada secuencia en el problema de interés, lo cual provoca que se deba generar completar tamaños con *zero-padding* para poder llevar a cabo dicha transformación.

Variados métodos computacionales han empleado estas técnicas combinadas con arquitecturas de *convolutional neural network* para desarrollar modelos predictivos en ingeniería de proteínas. Logrando predecir satisfactoriamente efectos termodinámicos, clasificación enzimática, entre otros, con rendimientos por sobre el 90 % de precisión.

Dentro de los principales métodos desarrollados se encuentra *ProtConv* [146], herramienta que utiliza el modelo pre entrenado *TAPE* [135] para representar numéricamente una secuencia de aminoácidos. Este vector es redimensionado para generar una imagen rectangular, la que se utiliza para predecir funciones proteicas utilizando arquitecturas de *deep learning*. Por otro lado, [185] construye matrices bidimensionales a partir de las propiedades de los aminoácidos que la componen, procesando su información previamente. Este método, además, se ha utilizado para evaluar interacciones proteína-proteína.

Por otro lado, se han implementado visualizaciones en espacios 20D (20 refiriéndose al número total de aminoácidos disponibles). La secuencia queda representada como "pasos" en un vector de 20 dimensiones, donde cada valor es representado como un punto de masa [116]. Además, se ha ampliado este enfoque considerando el momento de inercia tridimensional, con proyecciones a 2-D y 3-D para representar las distribuciones de aminoácidos [42]. El objetivo de estos métodos es realizar comparaciones de secuencias sin involucrar alineamientos, ya que resultan costosos computacionalmente.

Como una representación utilizando imágenes tridimensionales, se han realizado enfoques que consideran representar la proteína como un cilindro. Se procede a fijar los animo ácidos en un plano bidimensional, en forma de círculo, y utilizar el eje z para representar la secuencia con líneas interconectadas en la superficie del cilindro [186].

Para finalizar, existe el algoritmo *chaos game representation* (CGR), el cual ha sido utilizado desde la década de los 90 para representar secuencias genéticas a través de imágenes bidimensionales [79]. Últimamente, se ha ampliado este objetivo a proteínas, las cuales tienen un alfabeto de 20 aminoácidos, en lugar de 4 nucleótidos. CGR ha sido empleado exitosamente como método para clasificar funciones de proteínas en espacios reducidos de categorías[161].

Como método de clasificación de proteínas, durante los últimos años se han utilizado las redes neuronales de convolución tratando imágenes como entradas. En [145] genera una precisión del 85.3 % utilizando la arquitectura *LeNet-5*. Mientras, [44] utiliza la arquitectura *ResNet*, obteniendo un rendimiento del 95.03 %. Finalmente, con un mejor resultado, [89] obtiene una precisión del 96.9 % utilizando una arquitectura propia.

Sin embargo, a pesar de los enormes rendimientos, existen diferentes problemáticas que se deben considerar a la hora de aplicar estas metodologías.

1. La generalización de los modelos predictivos es compleja, si bien los sistemas predictivos logran una precisión superior al 80 %, la precisión y el *recall* son bajos (inferiores al 50 %).
2. No existen estrategias claras de representación de imágenes, razón por la cual, se deben explorar y combinar diferentes métodos para obtener los conjuntos matriciales.
3. La identificación de patrones puede inducir a falsos positivos, es decir, imágenes con un patrón muy similar. Pero, clasificados con distintas categorías, esto inducirá errores en los modelos predictivos y bajará el rendimiento de los modelos.

2.3. Aplicaciones de digital signal processing

El procesamiento digital de señales es una de las técnicas más utilizadas para comprender los fenómenos físicos de los espectros y sus características, teniendo aplicaciones en diferentes campos de investigación, como la biomedicina [39], biotecnología [156], entre otros. El enfoque de procesamiento digital en ingeniería de proteínas se basa en usar las transformadas de Fourier para obtener representaciones en el espacio de las frecuencias, estas representaciones facilitan emular el comportamiento estructural de los aminoácidos, dado que, en una

estructura, todos los residuos influyen sobre otro, mientras que en el dominio de las frecuencias, todo punto influye sobre el resto. A continuación, se exponen las principales temáticas asociadas a este tema, así como también los usos en ingeniería de proteínas. Este enfoque se detalla de manera más extensa debido a que es el foco de la metodología de representaciones de proteínas a desarrollar durante este capítulo.

2.4. Transformaciones de Fourier

Las transformadas de Fourier, corresponden a una transformación matemática que permite analizar una función definida en el espacio-tiempo, denominada señal, en sus frecuencias constituyentes. Como característica general, se genera una función definida en el espacio de frecuencias, representada por un valor complejo, en el cual, su módulo corresponde al valor de dicha frecuencia en la función inicial y su coeficiente, corresponde al desfase sinusoidal en la frecuencia [159].

Sea f una función definida en el espacio-tiempo, representando una señal, integrable Lebesgue, su transformada se define como $f : \mathbb{R} \to \mathbb{C}$, asociada a una frecuencia, denotada por $\hat{f}$, la cual se expresa como:

$$\hat{f}(\xi) = \int_{-\infty}^{+\infty} f(x) e^{-2\pi i x \xi} dx \tag{2.2}$$

Donde ξ corresponde a un número real y x representa al tiempo. A partir de 2.2, se puede definir la transformación inversa

$$f(x) = \int_{-\infty}^{+\infty} \hat{f}(\xi) e^{-2\pi i x \xi} d\xi \tag{2.3}$$

Tanto 2.2 y 2.3, corresponden a funciones con distribución continua. De manera similar, es posible definir las transformadas de Fourier y su inversa, en el espacio de distribuciones discretas, en donde, solo se considera un segmento muestral finito del conjunto de datos continuos para reconstruir el espectro de frecuencias [134]. Dado esto, la transformada de Fourier discreta se define como:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad \forall\ k \in [0, N-1] \tag{2.4}$$

Donde N representa a una secuencia de números complejos $x_0, \cdots, x_{N-1}$. Se define la transformada discreta inversa de Fourier como:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e \frac{2\pi i}{N} kn \quad \forall\ n \in [0, N-1] \tag{2.5}$$

Las transformadas de Fourier han sido utilizadas en diferentes campos de investigación, tales como: física, teoría de números, procesamiento de señales, propagación de ondas, óptica, etc. Siendo el análisis armónico, la rama matemática encargada de este tipo de estudios.

A partir de lo anterior, debido a los diferentes empleos que puede tener esta transformada, nace la necesidad de resolver de manera eficiente esta función, para ello nacen diferentes algoritmos, dentro de los cuales, el principal se conoce como Transformada rápida de Fourier (FFT, por sus siglas en inglés).

2.4.1. Transformada rápida de Fourier (FFT)

La transformada rápida de Fourier (FFT), es un algoritmo que permite encontrar solución a una DFT con una disminución en la complejidad. Esto es, al resolver el problema directamente desde la DFT, se presenta una complejidad $O(N^2)$, en cambio, al utilizar FFT, se obtiene una complejidad de $O(N \log N)$ [172].

La idea general del algoritmo fue propuesto por Cooley [36]. Dentro de sus particularidades, es que debido a la subdivisión en N transformadas de menor complejidad a resolver, donde N se compone de n_1 y n_2, se requiere que el conjunto de muestras, presente un tamaño del orden $2 \cdot 2^n$, es decir, una potencia de 2. A pesar de que dicho algoritmo es uno de los más comunes para la resolución de transformadas de Fourier, no es el único, siendo algunos: Prime-factor FFT [86], Bruun's FFT, Rader's FFT, Bluestein's FFT, y Hexagonal FFT [41].

Las definiciones matemáticas del proceso, se realizaron por Peter D. Welch en [172], explicando la formulación del problema y las demostraciones de la solución.

La división que se genera en el algoritmo FFT propuesto por Cooley [36], se basa en el uso de radix-2 DIT, esto es, la división de una DFT de tamaño N en dos DFT de tamaño $N/2$ de manera recursiva.

De manera general, se estiman las DFT de los pares e impares por separado (x_{2m} y x_{2m+1}, respectivamente), para luego combinarlas y estimar la DFT del espacio completo Debido a esta subdivisión recursiva en pares, se requiere un número de componentes en potencia de 2. Normalmente, se adicionan elementos para poder cumplir con dicha condición, comúnmente, se utiliza *zero-padding* [112], para satisfacerlo.

Matemáticamente, las DFT de los componentes pares e impares y su combinación se obtiene a partir de:

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N} 2mk} + \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N}(2m+1)k} \tag{2.6}$$

Donde el primer componente denota los elementos pares E_k y el segundo componente los impares O_k en la ecuación 2.6, respectivamente.

Un esquema representativo de los pasos a seguir en el algoritmo, la utilización del radix-2

DIT y cómo se obtienen las DFT para luego combinarlas se expone en la Figura 2.3.

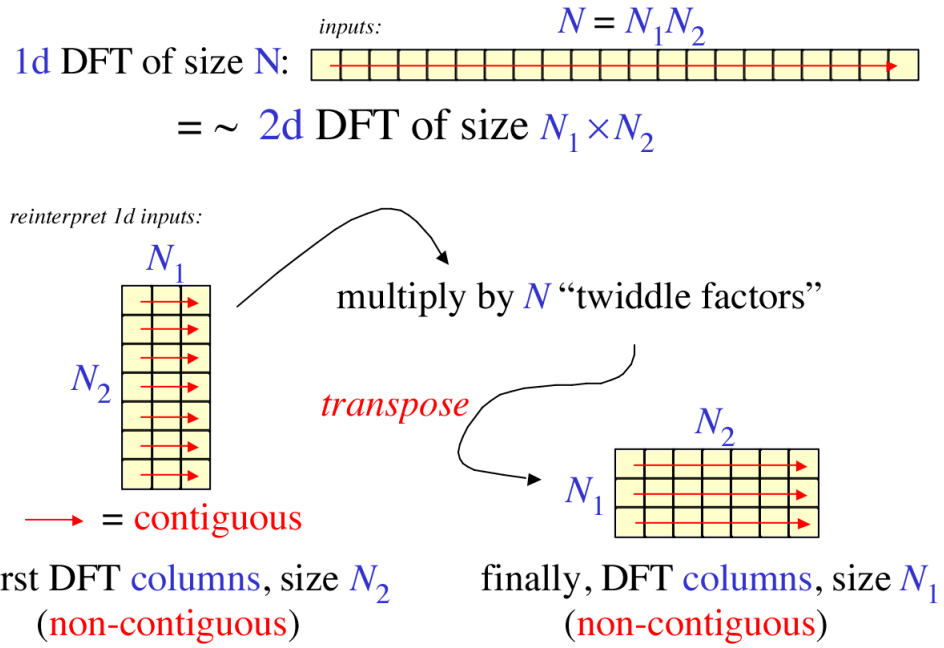


Figura 2.3: Esquema representativo de los pasos asociados al algoritmo FFT, desarrollado por Cooley [36]

2.4.2. Aplicaciones de las transformadas de Fourier en ingeniería de proteínas

Tal como se ha venido mencionando, las aplicaciones en ingeniería de proteínas para las transformadas de Fourier han permitido el desarrollo de diversas metodologías y estrategias. Los primeros estudios se enfocaron en el análisis de secuencias lineales de proteínas y DNA, con el fin de identificar residuos y posiciones claves correlacionados con peaks en el espectro de frecuencia [168, 40]. No obstante, esto resulta ser complejo a la hora de emplear FFT como método de resolución de las transformadas, ya que la solución inversa del problema no es factible, lo cual hace que los diferentes peaks en realidad los asocien a perfiles funcionales de proteínas. No obstante, esto último no ha sido ampliamente probado.

Estudios posteriores han aplicado la codificación por propiedades fisicoquímicas combinadas con las FFT para estudiar propensiones de moléculas orgánicas, relacionadas con toxicidad, actividad de antibióticos, carcinogénesis, entre otras propiedades [168, 38, 40].

Un enfoque más reciente es el uso de estrategias de modelos asociados al reconocimiento de resonancias (RRM, por sus siglas en inglés), los cuales han sido aplicados en diferentes problemas del área médica, reconocimiento de *hot spots*, entre otros, y, en particular, enfocados en el estudio de mutaciones en proteínas [38, 39, 40]. Un enfoque común, asociado al estudio de señales y modelos RRM, consiste en, a partir de la secuencia lineal de residuos, emplear la propiedad PEII (*Potential electron ion interaction*, por sus siglas en inglés) como método de codificación, posteriormente se aplica FFT como método de digitalización, y final-

mente se obtiene el espectro de frecuencia y se hacen los análisis de señales correspondientes [168, 38, 39, 40].

Estudios recientes se han centrado en aplicar esta estrategia de codificación para generar conjuntos de datos, los cuales, combinados con algoritmos de aprendizaje supervisado, permitan el diseño e implementación de modelos predictivos para diferentes tareas en ingeniería de proteínas [22, 107, 156]. Los modelos generados facilitan la predicción de tareas complejas en ingeniería de proteínas. Sin embargo, pese a sus grandes ventajas y usos, existen ciertas complicaciones que han de ser resueltas para poder aplicarlas de manera confiable en ingeniería de proteínas. Dentro de las cuales se pueden nombrar las siguientes:

1. Es necesario seleccionar las propiedades más informativas para poder emplearlas como inputs en los métodos de codificación y representación numérica de proteínas. Sin embargo, no existe un consenso con respecto a las propiedades más informativas.
2. Se requiere de métodos computacionales que faciliten el desarrollo de pseudo *autoencoders* de espectros de frecuencia, con el fin de poder emplearlo en métodos de diseño de mutaciones.
3. Se requiere de métodos y estrategias que faciliten el estudio del diseño de variantes y la evaluación de espacios latentes estadísticos, aplicando las representaciones de señales y perfilando espectros por función o actividad.

Contemplando las ventajas de las transformadas de Fourier y su relación con las propiedades estructurales e interacción entre los residuos que componen una secuencia, serán el foco de interés de este capítulo y la base para los principales desarrollos de este trabajo de tesis, contemplando la implementación de modelos predictivos, el uso para diseño de proteínas con propiedades deseables y la identificación de patrones y su aplicación como sistemas de transferencia de aprendizaje.

2.5. Principales problemáticas asociadas a la representación de proteínas

Con base en lo planteado durante este capítulo, se denota que existen diferentes técnicas, metodologías y estrategias que facilitan la representación numérica de secuencias de proteínas para su aplicación en desarrollo de sistemas predictivos. Cada una de las cuales, presenta ventajas y desventajas, que las hace interesantes de trabajar.

Si bien, el foco de este capítulo es el uso de las propiedades fisicoquímicas combinadas con transformadas de Fourier como estrategia de codificación de secuencias de proteínas, se trabajarán en diferentes problemáticas nombradas previamente, que abarcan a las diferentes metodologías de representación numérica. Dentro de estas propiedades destacan.

1. Identificar cuáles son las propiedades más representativas desde la base de datos *AAIndex*, con el fin de emplearlas como base para la codificación de proteínas.

2. Determinar los efectos de sobre ajuste, generalización y rendimiento de las propiedades combinadas con las FFT en comparación a los métodos clásicos de codificación y los basados en grafos e imágenes.
3. Evaluar la usabilidad de estas representaciones como método de clustering e identificación de patrones en familias de proteínas o conjuntos de proteínas no etiquetados.
4. La factibilidad de emplear estrategias computacionales como soporte de diseño de variantes con propiedades deseables.

Algunas de las problemáticas nombradas previamente, se asociarán con capítulos siguientes de esta tesis de doctorado. No obstante, se darán pequeñas muestras de usabilidad, debido al contexto de este capítulo. Sin embargo, cada estrategia será profundizada en su correspondiente capítulo.

2.6. Metodología

Con el fin de cumplir con los objetivos planteados y demostrar la hipótesis propuesta, se diseñó e implementó un conjunto de estrategias computacionales, las cuales se pueden dividir en los siguientes puntos descritos a continuación

2.6.1. Selección de propiedades desde AAIndex

El primer foco de desarrollo, se centró en la identificación de propiedades fisicoquímicas desde la base de datos *AAIndex* con el fin de emplearlas en sistemas de codificación o representación numérica de secuencias de proteínas.

La Figura 2.4 resumen las cuatro etapas desarrolladas para la identificación de las propiedades. Debido a que el principal foco es el desarrollo de grupos con sentido semántico, se emplean técnicas de *text mining*, principalmente *doc2vec* para su desarrollo.

Primero, se hace un procesamiento de las propiedades existentes en la base de datos *AAIndex*. 566 propiedades (descripciones y valores para los 20 aminoácidos canónicos) fueron descargados, los cuales se procesaron y dividieron en dos archivos con formato *.CSV con el fin de facilitar su manipulación, separando las descripciones de las propiedades de sus respectivos valores. Estos archivos tienen en común el código de la propiedad, con el fin de no perder la relación entre ambos elementos. Luego, se extrajeron todas las descripciones de las propiedades para poder someterlas a la siguiente etapa de la metodología.

En una segunda etapa, se combinaron las técnicas de *doc2vec* para el desarrollo de sistemas *autoencoders* con algoritmos de aprendizaje no supervisado para identificar grupos desde las descripciones. Todas las descripciones se someten al sistema de aprendizaje de *autoencoder*. Para ello, se implementan scripts basados en lenguaje de programación Python v3.9 y se empleó la librería *Gensim* para el entrenamiento de los *autoencoders*. Se varían diversos hiperparámetros de configuración en esta etapa, contemplando variaciones en las tasas de error,

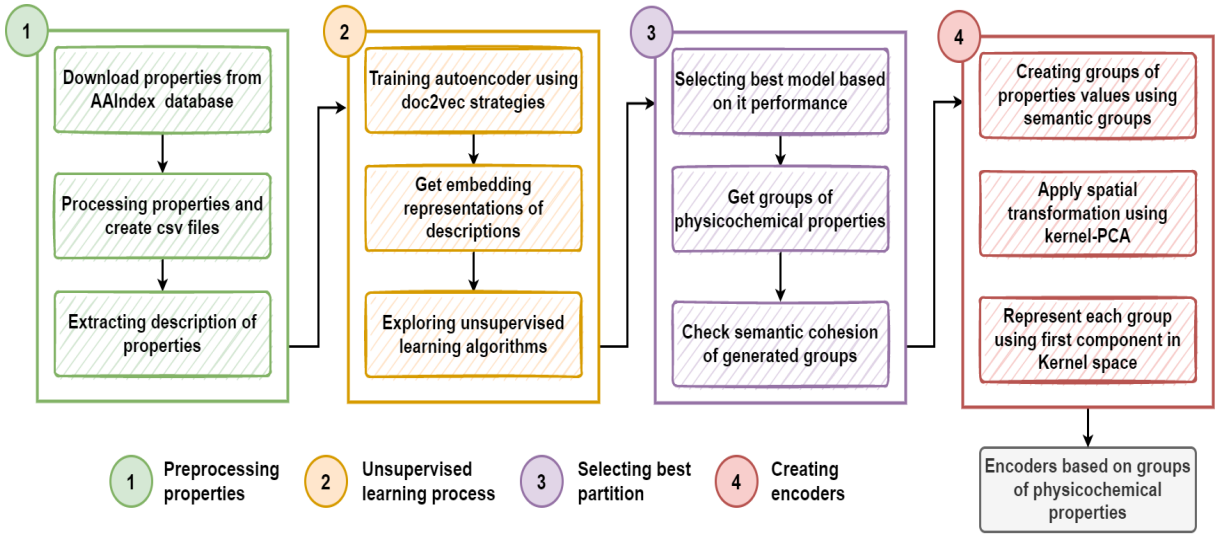


Figura 2.4: Resumen de las etapas desarrolladas para identificar las propiedades fisicoquímicas representativas

funciones de activación, error mínimo aceptable y la cantidad de épocas de entrenamiento. Además, debido a que el principal objetivo de un *autoencoder* es desarrollar un sistema de aprendizaje para generar vectores reales comprimidos (*embedding*) desde los textos, se varía el tamaño del vector desarrollado entre 2 y 1000. Junto con la exploración de la generación de los vectores numéricos, también se exploran diferentes algoritmos de aprendizaje no supervisado clásico, tales como K-Means, jerárquicos, *DBScan*, *affinity propagation*, *optics* y *mean shift*, entre los principales, variando también los diferentes hiperparámetros de configuración con respecto a cada algoritmo. El desempeño de las particiones generadas se midió con las métricas de *Calinski-Harabasz index* y Coeficiente de Siluetas.

En una tercera etapa, dada la exploración de generación de *embedding* combinadas con la exploración de identificación de particiones, se hizo un procesamiento estadístico con el fin de seleccionar las mejores combinaciones de desarrollo de *embedding* con la mejor forma de separar el conjunto de datos. Luego, se obtuvieron las descripciones de las propiedades fisicoquímicas agrupadas según los resultados de las particiones mediante algoritmos de aprendizaje no supervisado y se revisó la semántica de los grupos, esto es, que las descripciones de cada grupo fueran similares entre sí y que existieran diferencias con las de otros grupos generados y en el caso de existir discrepancias modificarlas manualmente, es decir, reclasificarlas a los grupos que corresponden.

Finalmente, la cuarta etapa de desarrollo de esta metodología de identificación de grupos de propiedades fisioquímicas con sentido semántico contempla la creación de los grupos de valores de propiedades. Esto es, para cada grupo se crea una matriz de $20 \times N$ donde 20 representa el número de residuos y N el número de propiedades para un grupo en particular. Luego, a cada una de las matrices generadas de manera independiente se le aplicó una estrategia de transformación espacial o reducción de dimensionalidad, en este caso, se aplicó Kernel-PCA, seleccionando como Kernel el *radial basis function*, ya que es el que más se ajusta a posteriores aplicaciones de FFT. Se descarta el uso de técnicas lineales debido a que requieren que los datos distribuyan normal para su aplicación, lo cual condicionaba

su uso, siendo necesario eliminar registros, pudiendo afectar negativamente a los resultados obtenidos. Finalmente, se tomó el primer componente en cada transformación espacial no lineal y se empleó para generar codificadores de los residuos.

De esta forma, se generaron n vectores numéricos de tamaño $k = 20$, los cuales representan los codificadores de secuencias amino acídicas, provenientes del procesamiento semántico de las propiedades termodinámicas y fisicoquímicas de la base de datos *AAIndex*.

2.6.2. Selección de casos de estudio y preparación de los conjuntos de datos

Con el fin de evaluar la versatilidad de los codificadores basados en propiedades fisicoquímicas semánticamente agrupadas, se diseñaron diferentes casos de estudio asociados a variadas tareas de ingeniería de proteínas. La Tabla 2.1 describe brevemente los conjuntos de datos, la cantidad de ejemplos y la tarea asociada, así como también el objetivo de su desarrollo.

Caso Estudio	Ejemplos	Tarea	Descripción
Clasificación DNA Binding protein	1027	Clasificar	Clasificación de proteínas de unión a DNA, generando un clasificador binario
Clasificación de Familias de enzimas	1827	Clasificador Múltiple	Clasificación de enzimas según familia, se consideran 6 grupos de familias.
Clasificación de función	3132	Clasificador Múltiple	Clasificación de función de proteínas en un grupo con mismo plegamiento
Clasificación de plegamiento	400	Clasificador Múltiple	Clasificación de plegamientos en grupo de proteínas con misma función

Tabla 2.1: Resumen de casos de estudios generados a modo de corroboración

Con respecto a la preparación de los conjuntos de datos, se plantea primero la codificación de las secuencias empleando los codificadores propuestos, seguidos de la aplicación de las transformadas de Fourier para representar las secuencias en el espacio de frecuencias.

Importante destacar, que la selección de casos de estudio se basó en los siguientes criterios.

1. Evaluar si los codificadores propuestos facilitan una representación eficiente para el entrenamiento de modelos predictivos en diferentes tareas de ingeniería de proteínas.
2. Estimar la factibilidad de reconocer plegamientos en un conjunto de proteínas clasificadas con la misma función o familia.
3. Estimar la factibilidad de reconocer funciones en un conjunto de proteínas con el mismo plegamiento.

En este capítulo, se dejan fuera los elementos asociados a la identificación de patrones y el uso de las transformadas de Fourier y sus representaciones para el diseño de secuencias y análisis de espacios latentes, debido a que serán focos de trabajo en los siguientes capítulos.

2.6.3. Comparaciones de rendimiento para tareas de ingeniería de proteínas

Con el fin de determinar si los codificadores propuestos juegan un rol importante a la hora de codificar secuencias y entrenar modelos predictivos, se implementaron diferentes codificaciones clásicas enfocadas en técnicas como *one hot* y *TAPE*. Se evaluó el efecto de las trasformadas de Fourier sobre las representaciones para determinar si existe una sinergia entre los codificadores y su transformación al espacio de señales.

Por otro lado, se aplicaron estrategias de *reshape* para obtener representaciones matriciales y comparar los modelos desarrollados contra los entrenados mediante técnicas de *convolutional neural network*. Se descartaron las representaciones por técnicas de grafos, debido principalmente a que el foco de este estudio es trabajar con proteínas que en su mayoría no tienen representación estructural, ya sea desde cristalografía de rayos X, resonancia magnética nuclear, o por modelos estructurales o complejos obtenidos con *AlphaFold* o *RoseTTAfold*.

2.6.4. Implementaciones y comentarios generales

Todos los desarrollos generados para llevar a cabo cada uno de los objetivos y metodologías propuestas se llevaron a cabo implementado scripts bajo lenguaje de programación Python v3.9 junto a sus diferentes librerías con el fin de facilitar la programación. Dentro de estas librerías se encuentran i) *DMAKit* [108] para aplicaciones del machine learning clásico, ii) *Tensorflow* [45] para aplicar *convolutional neural network* (CNN) y iii) Gensim [139] para trabajar con sistemas *autoencoders* y desarrollo de *embedding*. Los componentes de desarrollo se empaquetaron en un ambiente *conda* para facilitar la reproducibilidad de los resultados obtenidos y compartir las estrategias desarrolladas.

2.7. Resultados y discusiones

2.7.1. Identificación de grupos semánticos de propiedades fisicoquímicas

Empleando los registros de la base de datos *AAIndex* [82] junto a la combinación de estrategias de *doc2vec* y algoritmos de aprendizaje no supervisado, se identificaron ocho grupos semánticos de propiedades fisicoquímicas, los cuales se visualizan en la Figura 2.5. Estos grupos representan diferentes propiedades generales de los aminoácidos, tales como propiedades estructurales, termodinámicas e índices, pudiendo ser utilizados como métodos de selección de propiedades fisicoquímicas para facilitar la codificación de secuencias.

En total, cerca de 1 millón de formas de particionar el conjunto de descripciones de propiedades fueron exploradas, combinando tanto la forma de generar los *embedding* como los algoritmos e hiperparámetros aplicados para particionar el *conjunto* de datos.

Los ocho grupos obtenidos se lograron a partir del entrenamiento de *autoencoders* con hiperparámetros de 500 épocas, un valor de $\alpha = 0{,}025$ y un tamaño del *embedding* de 2, por otro lado, la partición se generó aplicando el algoritmo de $k-$means con un $k = 8$, el desempeño logrado de esta estrategia fue de un *Calinski-Harabasz index* de **1532.36** y un coeficiente de siluetas de **0.43**, siendo la mejor forma de particionar el conjunto de descripciones de propiedades dentro de las exploradas.

Por último, se verificó la semántica de cada grupo, evaluando que presentaran los mismos contextos, palabras específicas o tópicos, a raíz de lo cual solo **17** descripciones fueron reclasificadas debido a que fueron agrupadas en una primera instancia el grupo de los *."other indexes"* y pertenecían al grupo de las propiedades *alpha structure* y *beta structure*.

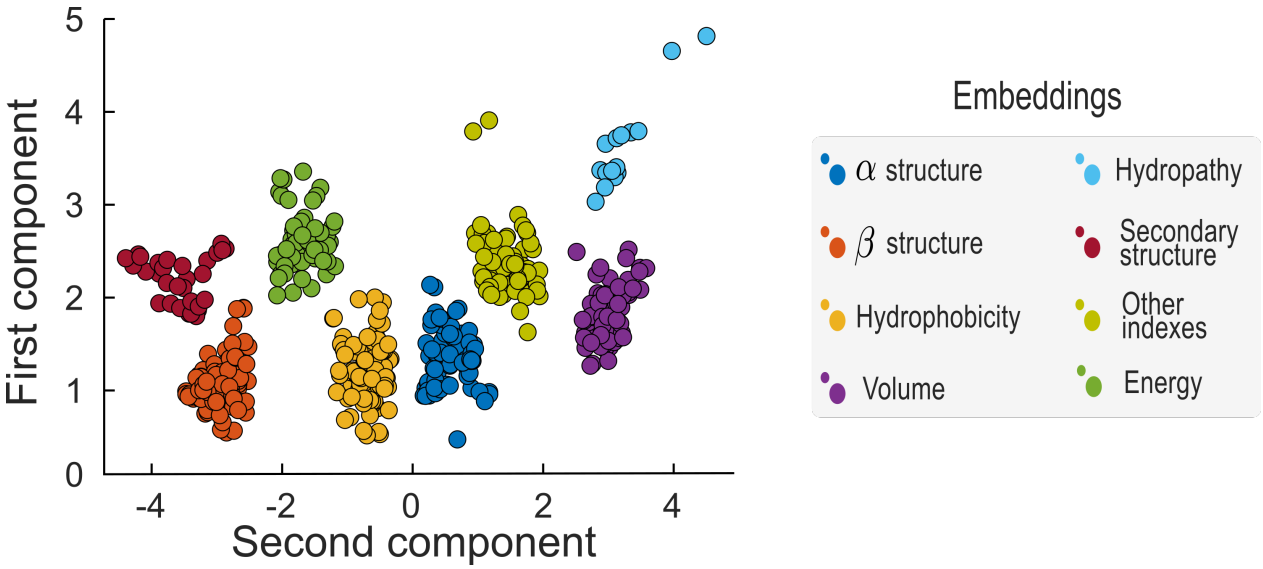


Figura 2.5: Representación de *scatter plot* de los ocho grupos de propiedades semánticas identificados empleando estrategias de *doc2vec* combinadas con aprendizaje no supervisado

Uno de los principales motivos y a la vez ventajas de implementar una estrategia basada en *doc2vec* es la semanticidad que se genera al separar las propiedades por sus descripciones, lo cual facilita una visualización simple de los principales contextos o tópicos existentes en cada grupo. Además, al aplicar algoritmos de aprendizaje no supervisado sobre los valores de las propiedades, las particiones generadas no aseguran una separación en grupos con sentido semántico. De esta forma, la metodología de clustering semántico, propuesta en este trabajo, se asegura que la selección al azar de cualquier integrante de un grupo en particular, tendrá el mismo significado o descripción. Lo cual no se asegura con grupos desarrollados a partir de los valores numéricos, ya que, evidentemente, estos grupos tendrían un sentido numérico entre ellos o una correlación. No obstante, dicha correlación no está relacionada con la descripción de las propiedades, sino que con los valores de las propiedades en sí.

Se empleó la exploración de algoritmos de aprendizaje no supervisado sobre los valores de las propiedades, logrando el mejor desempeño una partición con $k-$means=2 con rendimientos de *Calinski-Harabasz index* de **1527.81** y un coeficiente de siluetas de **0.87**. Si bien, los resultados muestran excelentes separaciones según los criterios de rendimiento, a la hora de analizar los grupos, no solo no existe una relación de las descripciones entre los grupos, sino

que también se generaron divisiones desbalanceadas, es decir, la mayoría de los ejemplos se encuentran en un único grupo.

Finalmente, al forzar una partición de **8** grupos sobre el conjunto de valores de propiedades se obtienen rendimientos de **614.25** de *Calinski-Harabasz index* y **0.50** de coeficiente de siluetas, los cuales no solo son más bajos que los obtenidos por la metodología propuesta, sino que también no se logran grupos semánticos.

Ya con los grupos de descripciones generados y corregidos, estos fueron utilizados para generar ocho conjuntos de datos con los valores de las propiedades para cada aminoácido. Aplicando transformaciones espaciales basadas en kernel-PCA, se transformaron los valores de las propiedades para cada grupo independiente y se evaluó la varianza explicada para cada caso. En todos los grupos, la varianza explicada fue superior al **85 %**, contemplando esto, se propone emplear como codificador el primer componente de cada grupo semántico de propiedades generados. Un punto importante a mencionar, destaca en el hecho de que al usar este componente, en realidad se emplea un número asociado a un vector que representa combinaciones no lineales que maximizan la varianza para el espacio analizado, lo cual toma la ventaja de que en realidad es una representación general de todos los miembros de un grupo en específico. Los codificadores son expuestos en la Tabla 2.2.

Residue	Alpha structure	Beta structure	Hydrophobicity	Volume	Energy	Hydropathy	Secondary structure	Other indexes
A	290.41	71.85	6.25	44.65	-107.79	15.33	56.16	92.92
R	172.57	-6.96	84.09	200.15	51.15	172.36	1.44	-37.39
N	-38.37	-90.14	-21.73	-191.18	73.94	-259.13	-54.69	-77.74
D	159.43	-56.58	-28.96	-232.26	55.36	-216.01	-29.38	-7.42
C	-4.24	15.67	-34.88	-156.21	-54.19	-242.01	10.07	40.04
Q	-268.55	-32.61	38.46	179.88	31.44	145.73	-15.43	-45.52
E	-0.02	21.03	-21.48	-170.44	-49.97	8.11	20.20	50.74
G	-104.49	-62.33	53.16	250.66	92.25	256.52	-39.89	-95.41
H	-159.87	31.27	-69.67	194.47	-39.54	455.61	34.12	43.37
I	-34.08	164.64	-54.85	-88.56	-48.44	-274.76	25.05	52.40
L	-91.11	-16.38	-64.98	-201.08	7.56	-257.27	-10.20	4.27
K	195.59	54.45	-52.92	-118.84	-109.99	-136.28	55.31	85.66
M	21.94	-18.77	-26.70	-227.61	-7.39	-139.71	-19.45	16.04
F	88.02	21.61	-21.46	-78.96	-56.97	80.68	30.31	46.42
P	317.10	115.37	-22.23	-44.80	-157.63	-126.45	95.69	136.09
S	-314.20	-106.56	61.31	221.12	174.08	248.05	-85.57	-122.66
T	-252.51	-23.99	13.72	-3.30	17.50	-153.13	-25.56	-31.46
W	-118.15	-76.02	88.28	34.80	105.47	19.24	-59.91	-124.49
Y	-10.20	-15.49	40.85	203.07	36.61	171.61	-4.25	-33.07
V	150.75	9.929	33.77	184.45	-13.45	231.50	15.99	7.21

Tabla 2.2: Codificadores de aminoácidos generados desde los grupos semánticos de propiedades fisicoquímicas

Un punto de interés en este approach es la selección de métodos no lineales para hacer las transformaciones espaciales, lo cual es debido principalmente a que si se aplican métodos lineales convencionales para transformar espacios como PCA, se requiere que las propiedades de cada grupo tengan una distribución normal, lo cual implicaría remover propiedades que no cumplen con dicha característica, lo cual conllevaría a rearmar los grupos de descripciones, pudiendo afectar negativamente los resultados obtenidos debido a la falta de información para aplicar métodos basados en *doc2vec*.

2.7.2. La combinación de FFT y codificadores de propiedades fisicoquímicas semánticas mejoran el rendimiento de modelos predictivos

Se emplearon los codificadores semánticos identificados con el fin de entrenar modelos predictivos basados en algoritmos de *random forest* para diferentes tareas de ingeniería de proteínas. Tales como, la clasificación de proteínas de unión a DNA, plegamiento y función de proteínas. Así como también, clasificación de familias de enzimas. Se comparó los resultados obtenidos con métodos clásicos de codificación como *one hot* [20] y con enfoques basados en *natural language processing* (NLP) con la herramienta TAPE [135]. Importante mencionar, que con el fin de brindar peso estadístico a los resultados mostrados, los análisis se repitieron 1000 veces y en todos los resultados expuestos se visualiza el promedio y una barra de error.

En la Figura 2.6 se muestra el desempeño de los modelos entrenados empleando los codificadores generados en este trabajo de doctorado para los diferentes problemas propuestos en los casos de estudio, descritos en la tabla 2.2. Los modelos entrenados se logran aplicando el algoritmo de *random forest*, cuya selección se basa meramente en una elección aleatoria, aplicando los mismos hiperparámetros de configuración en cada caso, con el fin de poder hacer comparativas en los diferentes codificadores y a su vez en el efecto de la aplicación del FFT, no evaluando el efecto del algoritmo y su forma de trabajar.

En la mayoría de los casos estudiados, al menos uno de los codificadores propuestos presentan un desempeño mayor a lo obtenido, aplicando estrategias como *one hot* y *embedding* (en al menos una iteración del proceso). No obstante, de manera general, no existe una diferencia significativa entre el uso de los codificadores generados basados en las propiedades fisicoquímicas y los métodos de codificación basados en *embbeding* y *one hot*. El hecho de que en algunos casos sea mejor el desempeño de los codificadores propuestos en este trabajo denota que una identificación y uso procesado de propiedades fisicoquímicas incrementa el desempeño de los modelos predictivos. No obstante, esto va de la mano a cómo se comportan los ejemplos en la división del proceso de entrenamiento, de tal forma, que se descartan diferencias significativas.

Por otro lado, se comparó los desempeños en la etapa de entrenamiento y de validación de los modelos generados para determinar la existencia de sobre ajuste en cada caso de aplicación. La Figura 2.7 muestra los resultados obtenidos.

Tal como se aprecia en la Figura 2.7, a medida que los valores de las tasas estimadas sean cercanas a 1 implica que no existe sobre ajuste o es mínimo, en caso contrario, para valores más alejados a 1, existe un mayor sobre ajuste, ya sea ajustando en el proceso de entrenamiento (valores mayores a 1) o en el proceso de validación (valores menores a 1).

En este caso, la mayoría de los modelos presenta sobre ajuste en la etapa de validación. No obstante, casos más drásticos como el entrenamiento por *one hot* para clasificación de plegamientos se expone un mayor índice. Por otro lado, en el caso de clasificación de proteínas de interacción a DNA, entrenados con codificación de propiedad fisicoquímica *Other Indexes*, los modelos presentan un sobre ajuste notorio en comparación al resto de sus pares. Esto es esperable debido al significado y el contenido de dicha variable, ya que, simplemente son

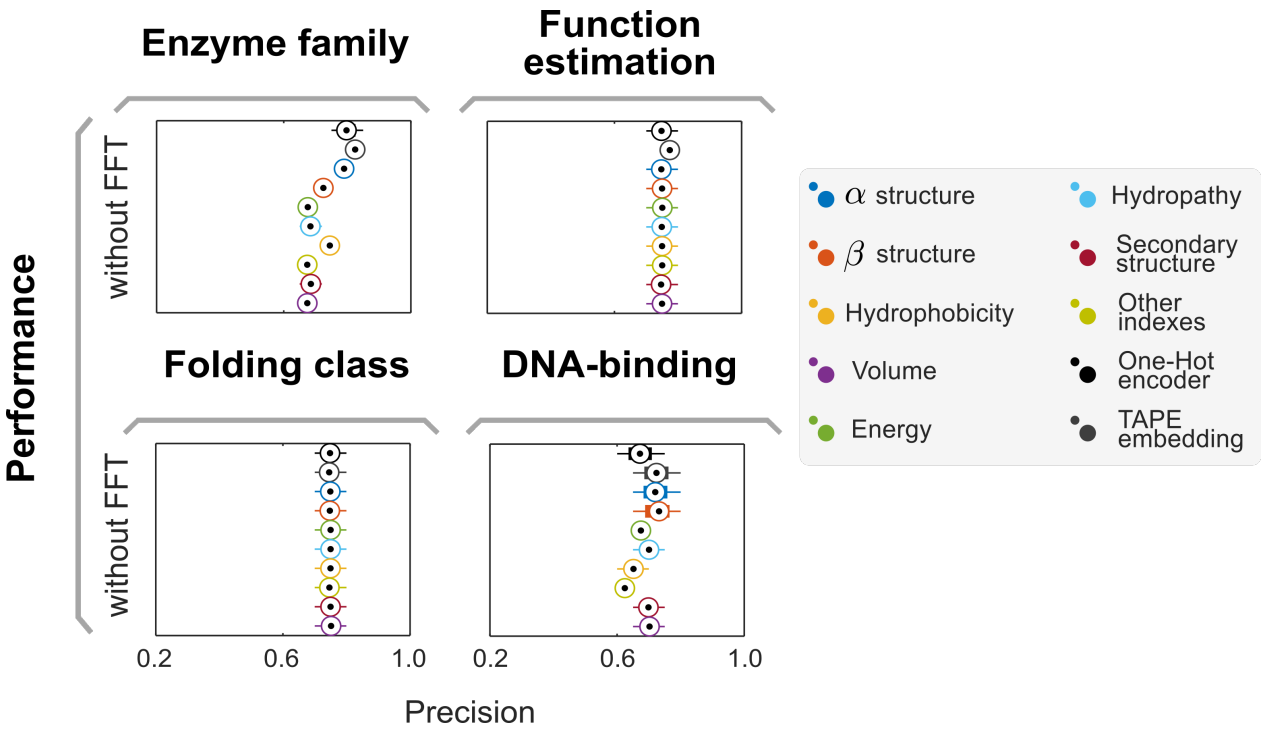


Figura 2.6: Medidas de desempeño obtenidas para los diferentes casos de estudio comparando las representaciones mediante los codificadores propuestos y representaciones clásicas.

índices de diferentes medidas o experimentos, lo cual no necesariamente asegura que tengan una relación más que una semanticidad entre las descripciones del grupo.

En la mayoría de los casos, con excepción de los nombrados, no existe una diferencia clara entre los modelos entrenados por los codificadores propuestos en comparación a las estrategias clásicas, tanto a nivel desempeño, en donde los resultados fueron un mejor desempeño para los codificadores semánticos, como en las tasas de sobre ajuste. Esto es, visualmente, se logran mejores desempeños con los codificadores de propiedades fisicoquímicas. Pero, al evaluar el sobre ajuste, no existen tendencias claras. Incluso, existen algunas propiedades que presentan valores de sobre ajuste elevados para todas las tareas de clasificación entrenadas, razón por la cual, no existen pruebas fehacientes de que se presenta una mejora en desarrollo de modelos con los codificadores planteados.

En vista de lo anterior y en concordancia de los objetivos propuestos y la hipótesis planteada, se procede a estudiar el efecto sobre las codificaciones que produce la aplicación de las transformadas de Fourier. Primero, se evaluó el desempeño de los modelos. La Figura 2.8 muestra los resultados obtenidos al aplicar transformadas de Fourier a las actuales codificaciones. Como se observa, existe una diferencia notoria entre los codificadores basados en *one hot* y *embedding* cuando se le aplica el FFT en comparación a los codificadores propuestos en este trabajo, denotando un efecto de sinergia entre los codificadores propuestos y la transformación en el espacio de señales para la representación de las secuencias de proteínas.

La sinergia identificada entre los codificadores semánticos y las transformadas de Fourier no solo beneficia el rendimiento de modelos predictivos, tal como se pudo observar en la Figura 2.8. Si no que también se aprecia en la disminución de las tasas de sobre ajuste estimadas

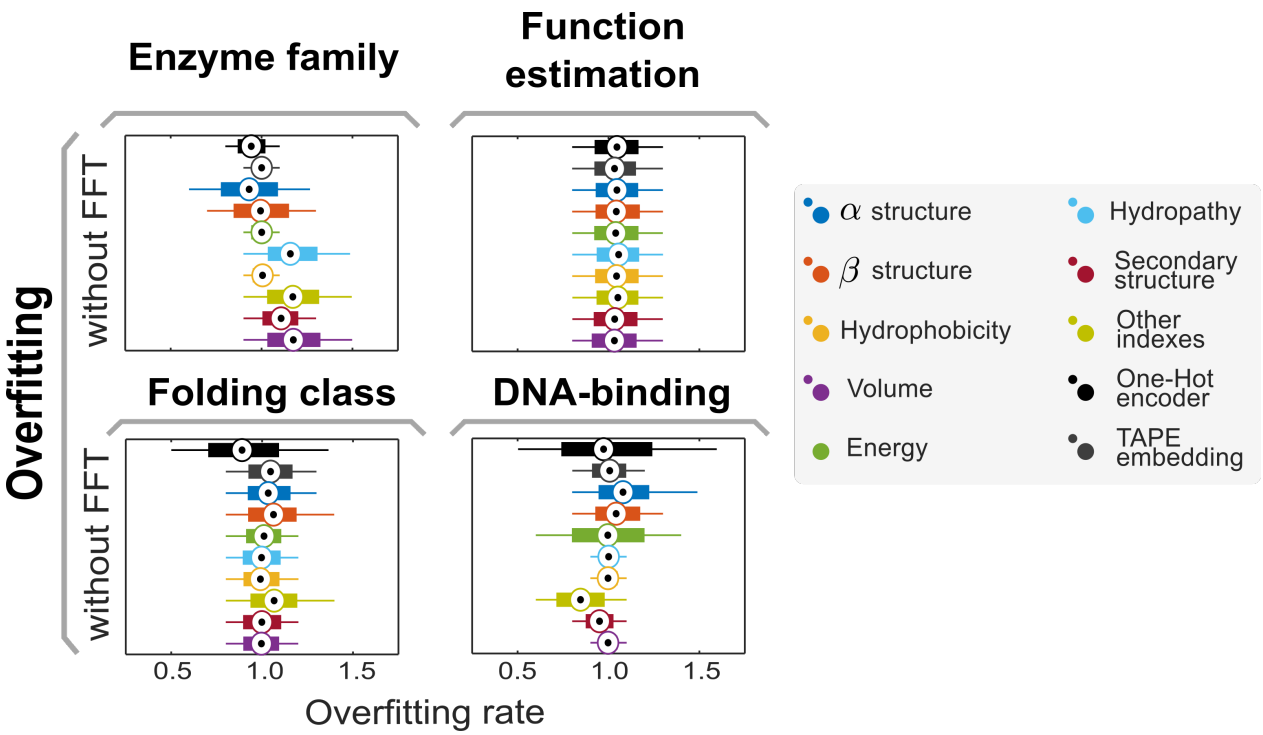


Figura 2.7: Tasas de sobre ajuste estimadas entre los desempeños de validación y entrenamiento para las diferentes pruebas desarrolladas.

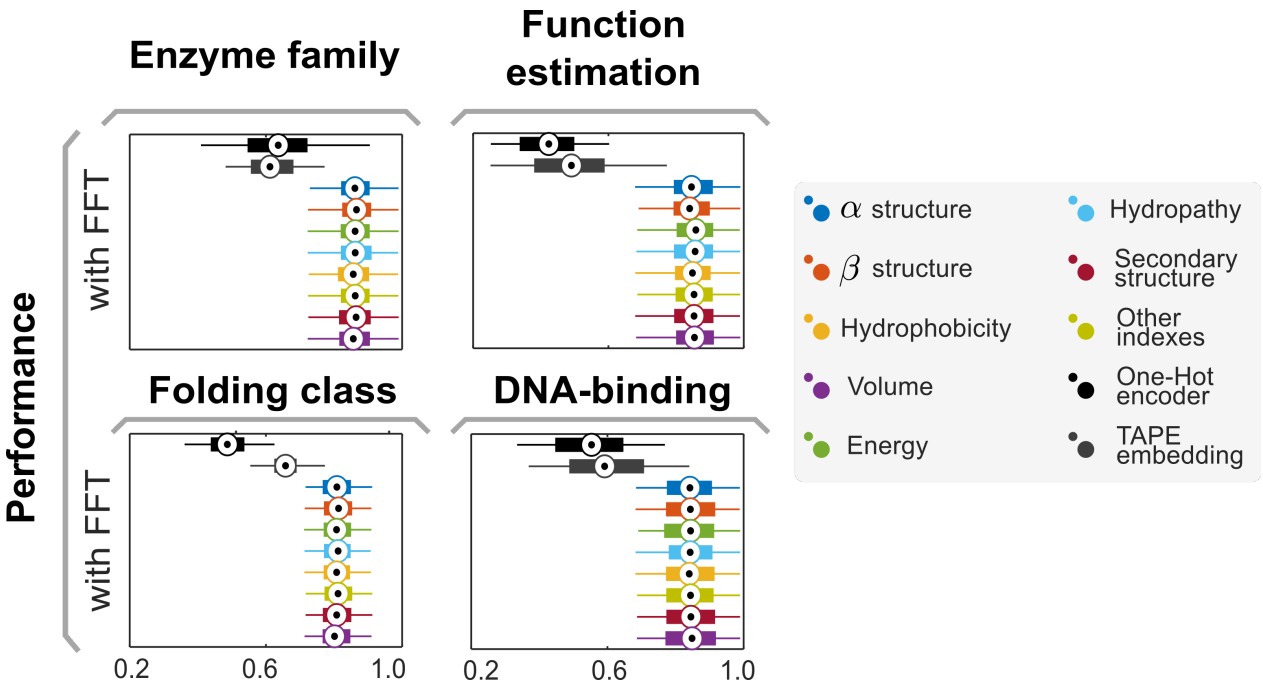


Figura 2.8: Medidas de desempeño obtenidas para los diferentes casos de estudio combinando las codificaciones con las transformadas de Fourier.

al comparar los desempeños de los modelos en las etapas de validación y entrenamiento, tal como se observa en la Figura 2.9.

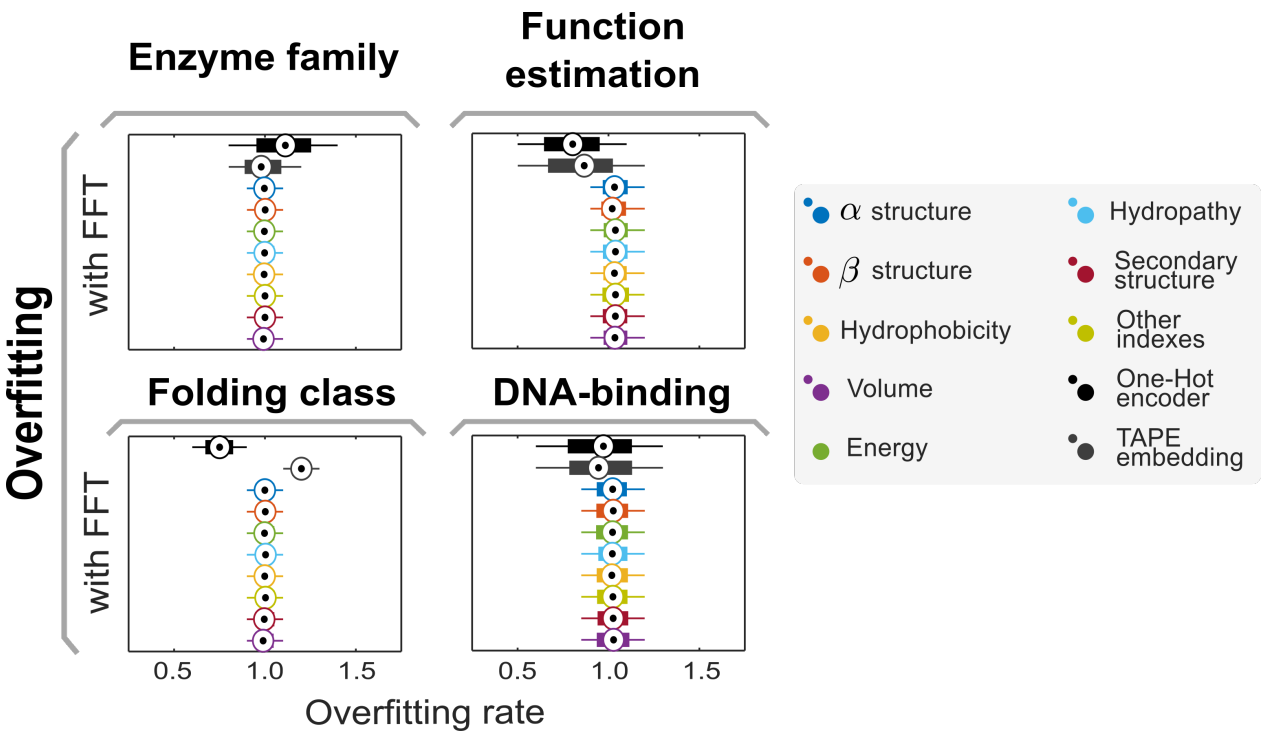


Figura 2.9: Tasas de sobre ajuste estimadas entre los desempeños de validación y entrenamiento para los modelos generados, combinando las representaciones numéricas con transformaciones en el espacio de señales.

Notablemente, las tasas de desempeño se ven favorecidas a la hora de aplicar representaciones en el espacio de señales para los codificadores propuestos con respecto a las diferentes tareas abordadas en este trabajo de tesis. A diferencia de los clasificadores generados por *one hot* y *Tape*, a los cuales, al aplicarles transformadas de Fourier, incrementan notoriamente su sobre ajuste, ya sea en la etapa de entrenamiento o en la etapa de validación, respectivamente. Mientras que para los codificadores propuestos combinados con las FFT las tasas de sobre ajuste se encuentran en un rango entre 0.8 y 1.2, lo cual demuestra que para los casos estudiados el sobre ajuste es mínimo.

Importante mencionar, que los mejores desempeños observados para cada caso (2.8 no corresponden a una propiedad en específico, Por ejemplo, para los casos de clasificación de proteínas de interacción con DNA, los mejores rendimientos se obtienen con las propiedades *Alpha-structure*, *Hydrophobicity*, y *Energy*, mientras que en clasificación de tipos de plegamientos de proteínas se logran con las propiedades relacionadas con el volumen. Con base en esto, se puede inferir que si bien se logran mejores desempeños al codificar con los codificadores propuestos en este trabajo combinados con FFT, en comparación con las estrategias basadas en *one hot* y *embedding*, en inclusive con las mismas propiedades fisicoquímicas, no existe una evidencia clara que haga denotar cuál es la única o mejor selección de una propiedad en específico.

Por otro lado, el efecto sinérgico que se produce entre los codificadores semánticos y sus representaciones en el espacio de señales se pueden explicar principalmente al efecto que provoca las transformadas de Fourier. Esto es, dado que en el espacio de señales existe una dependencia de cada punto con respecto al resto en el mismo espacio, esto emula los

comportamientos de los aminoácidos en las estructuras de las proteínas [39], lo cual, puede inducir a un mayor nivel de información entre los puntos ejemplos de una misma clase, lo cual facilitaría un mejor perfilamiento o caracterización de las clases en un conjunto de datos, en particular, para los casos de estudio que se han mostrado en este trabajo de doctorado.

Con el fin de evaluar si se logran perfilamientos eficientes, en las siguientes secciones se analizan cómo los codificadores semánticos combinados con las transformadas permiten reconocer o diferenciar plegamientos y funcionalidades en conjuntos de proteínas.

2.7.3. Reconocimiento de patrones visuales en plegamientos y funciones enzimáticas

Dos preguntas fueron planteadas previamente relacionadas con la representación numérica de secuencias. La primera se basa en la facilidad con la que una representación permite identificar patrones de plegamientos en funciones de proteínas, mientras que la segunda es cómo las representaciones facilitan el reconocimiento de funciones en proteínas con el mismo plegamiento.

Para ello, debido a los rendimientos obtenidos por los modelos predictivos entrenados empleando las codificaciones semánticas propuestas y combinadas con las transformaciones al espacio de señales y empleando la propiedad relacionada a las estructuras secundarias, se analizaron los perfiles de Fourier para cada caso y se compararon con el fin de evaluar si existe una respuesta a las problemáticas planteadas.

La Figura 2.10 A muestra los perfiles de Fourier (espectros de frecuencia) para las enzimas con función hidrolasas y ligasas. Como se aprecia en la Figura 2.10 A, existe una diferencia clara entre cada uno de los espectros promedios representados, lo cual se observa tanto en amplitud como en la diferencia de los peaks. Por otro lado, se hizo una división del conjunto de datos, dividiendo cada una de las enzimas de una función en específico en sus respectivos tipos de plegamiento, lo cual se observa en la Figura 2.10 B y Figura 2.10 C, para las enzimas hidrolasas y ligasas, respectivamente. Se observó en ambos casos una diferencia en los perfiles de plegamiento para cada una de las funciones enzimáticas analizadas. De esta forma, se puede mencionar, que las transformadas de Fourier combinadas con los grupos semánticos de propiedades, en este caso, la propiedad *Secondary-structure* facilita la identificación de perfiles que no solo separan de manera eficiente en función enzimática, sino que también ayuda al reconocimiento de perfiles de grupos proteínas por plegamientos.

Al analizar el caso inverso, es decir, clasificación de plegamientos y posterior análisis de funciones enzimáticas, los resultados son similares. La Figura 2.10 D, Figura muestra los perfiles para los plegamientos estudiados, existiendo notorias diferencias visuales entre los plegamientos α y β. Lo cual, al analizarlos individualmente con funciones específicas, en este caso isomerasas y óxido reductasas, se logra la identificación de diferencias entre ambos perfiles para los plegamientos individuales analizados (Figura 2.10 E y Figura 2.10 F).

Todos los análisis visuales de espectros analizados son concordantes con los resultados del desempeño obtenido para los modelos predictivos entrenados, demostrando la sinergia propuesta al combinar los codificadores de grupos semánticos con las representaciones en

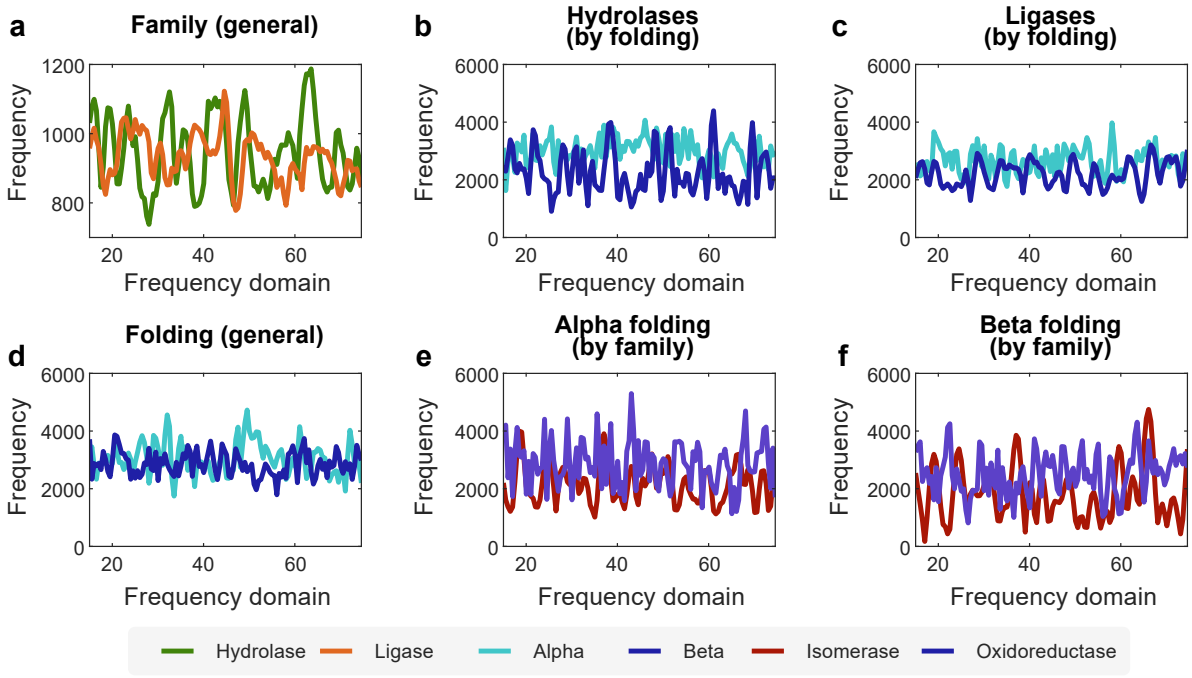


Figura 2.10: Evaluación de espectros de frecuencia como patrones de reconocimiento de secuencias en plegamiento y función enzimática

el espacio de señales. Esto último es interesante debido no solo a las aplicaciones en el entrenamiento de modelos predictivos, sino que también lo es a la hora de desarrollar sistemas de clustering o reconocimiento de patrones, así como también estrategias de diseño de nuevas variantes con propiedades deseables.

2.8. Conclusiones y comentarios generales

Con base en las estrategias de identificación de propiedades con sentido semántico y las aplicaciones de transformadas de Fourier expuestas en las secciones anteriores, es posible concluir algunos puntos de interés, los cuales se exponen a continuación.

1. La combinación de estrategias de text mining y algoritmos de aprendizaje no supervisado permiten la identificación de grupos de propiedades fisicoquímicas con sentido semántico, lo cual no es factible en un 100 % con las aplicaciones clásicas de algoritmos de clustering sobre los valores de las propiedades. De esta forma, se generaron ocho grupos de propiedades con sentido semántico.

2. Al aplicar estrategias de transformaciones espaciales no lineales, en este caso Kernel-PCA, sobre cada grupo, se pueden identificar codificadores que son representados a partir de componentes, las cuales corresponden a combinaciones no lineales de cada miembro de grupo. Para cada caso, la primera componente logra un mínimo de un 85 % de explicación de varianza, razón por lo cual, se propone como codificador. De

esta forma, se tiene una matriz de tamaño 20×8 donde 20 corresponde al número de residuos y 8 a la primera componente obtenida por Kernel-PCA de cada grupo identificado.

3. Los codificadores semánticos generados al usarlos en entrenamiento con técnicas de *machine learning*, presentan desempeño similar a las métricas obtenidas por representaciones clásicas. Sin embargo, exhiben un mayor sobre ajuste cuando se comparan con las codificaciones clásicas.

4. Los codificadores semánticos identificados combinados con transformaciones en el espacio de señales facilita el aprendizaje de modelos predictivos, denotando un incremento en el rendimiento y una disminución en el sobre ajuste, lo cual beneficia la generalización de comportamientos, no observable en las codificaciones estándares, en las que, al contrario, su desempeño disminuye.

5. No se logra identificar claramente cuáles son las propiedades más relevantes, dado que no existe un punto en común en todos los casos estudiados que indique con claridad que una propiedad en específico logra mejores resultados, razón por la cual, se recomienda explorar todas las propiedades propuestas, e inclusive combinarlas para obtener diferentes puntos de vista que no se identifiquen individualmente.

6. La combinación de las FFT con los codificadores semánticos facilita la identificación y perfilamiento de plegamientos y funciones enzimáticas. Además, a la hora de trabajar con plegamientos, permite identificar diferentes funciones enzimáticas, mientras que al evaluar funciones, facilita la identificación de plegamientos, demostrando capacidad de separar y perfilar grupos de secuencias con propiedades en común.

7. Se propone que la combinación planteada facilitaría no solo el entrenamiento de modelos predictivos, sino que también la aplicación de técnicas de clustering para la identificación de patrones y su uso en técnicas o estrategias de diseño de secuencias con propiedades deseables.

Gracias a los hallazgos y metodologías planteadas en este proyecto, se ha logrado no solo reconocer que existe una sinergia entre los grupos semánticos y las transformadas de Fourier, sino que también se plantea su uso a aplicaciones de identificación de patrones y diseño de variantes, los cuales serán abordados en los siguientes capítulos.

Capítulo 3

Métodos de ensamble para mejorar el rendimiento de modelos predictivos

Diseñar e implementar modelos predictivos se ha convertido en una de las tareas recurrentes en ingeniería de proteínas desde las últimas tres décadas. Tal como se pudo observar en el capítulo anterior, las estrategias de codificación son variadas y representan un punto importante a la hora de entrenar modelos basados en técnicas de machine learning, debido principalmente a que si se mejora la representación de los ejemplos, facilitaría el aprendizaje de los algoritmos de aprendizaje supervisado.

Por otro lado, existen otros puntos de interés que son interesantes de destacar a la hora de entrenar y desarrollar un modelo predictivo. Primero, los modelos generados deben ser generalizables, esto es, que tengan la capacidad de actuar indistintamente del conjunto de datos empleados para entrenar y validar, en otras palabras, que tenga un sobre ajuste mínimo hacia esos conjuntos. Por otro lado, se encuentra el poder predictivo de un modelo, esto es, cómo se asegura el poder predictivo de un modelo?, los rendimientos informados son suficientes para ello?

Además de los puntos nombrados, se destaca que si bien existe un protocolo común establecido para entrenar modelos, el cual se basa principalmente en la selección de algoritmos y su hiperparametrización, muchas veces no se exploran otras alternativas de algoritmos y solo se centran en abordar ciertos hiperparámetros que son de interés, no contemplando la combinación de varios modelos en un único sistema predictivo que permita generar respuestas a partir del uso de la combinación de las respuestas de los modelos independientes, lo cual se ha visto que mejora el rendimiento de modelos, tal como ocurre en los sistemas de ensamble como Bagging o Boosting. Así como también, la optimización de hiperparámetros de los modelos guiada por métodos de heurística como algoritmos genéticos. Por otro lado, los recientes avances del Deep Learning han fomentado su aplicación en variadas tareas de ingeniería de proteínas. Sin embargo, problemas relacionados con el sobre ajuste y a la falta de generalización son comunes en este tipo de aplicaciones.

Con base en lo anterior, este capítulo se centrará en la exploración de algoritmos de aprendizaje supervisado y sus hiperparámetros optimizados por sistemas de heurísticas basadas en algoritmos genéticos, así como también la combinación de modelos para la elaboración

de un sistema predictivo que mejore el rendimiento de los modelos individuales. De esta forma, primero se diseñó e implementó un modelo de optimización para identificar los mejores parámetros en un sistema de aprendizaje supervisado, luego, se propuso una estrategia de desarrollo de modelos predictivos ensamblados para maximizar el desempeño de los modelos individuales, emulando un sistema de bagging o boosting.

Las dos propuestas mencionadas se evaluaron en diferentes tareas de ingeniería de proteínas, contemplando como input diferentes representaciones numéricas clásicas, así como también el uso de los codificadores semánticos combinados con transformadas de Fourier desarrollados y validados en el capítulo anterior de este escrito de tesis de doctorado. En la mayoría de los casos, el ensamble de modelos predictivos logra un mejor rendimiento que los modelos individuales. No obstante, no se logra una significancia estadística robusta para afirmar este supuesto. Sin embargo, a la hora de combinar no solo los modelos, sino que las representaciones numéricas generadas con los codificadores semánticos desarrolladas en el espacio de las señales, se logra un aumento en el rendimiento del sistema predictivo, a partir de lo cual se propone que la combinación de diferentes puntos de vista de la descripción de una proteína mejora el rendimiento de los modelos predictivos y su generalización.

A continuación, se describen las metodologías actuales de entrenamiento de modelos predictivos para ingeniería de proteínas, basados en el estado de arte reciente, así como también, los principales hallazgos, las problemáticas analizadas, además de la metodología, los resultados y discusiones sobre las propuestas generadas.

3.1. Estrategias de entrenamiento de modelos predictivos

Las estrategias clásicas de entrenamiento de modelos predictivos en ingeniería de proteínas pueden dividirse en: i) Representación numérica de las secuencias, ii) Entrenamiento de modelo predictivo siguiendo estrategia de validación. iii) Evaluación del rendimiento y sobre ajuste del modelo [107].

Tal como se puedo apreciar en el capítulo anterior, la etapa de representaciones numéricas juega un rol fundamental a la hora de entrenar modelos predictivos. Sin embargo, no solo la representación juega un rol importante, sino que también la selección del algoritmo de aprendizaje y sus hiperparámetros de configuración [109].

La selección del algoritmo y los hiperparámetros normalmente se basa en exploraciones de ellos en métodos de combinación, donde se evalúan diferentes combinaciones de algoritmos e hiperparámetros y se selecciona según desempeño. En este sentido, uno de los puntos interesantes que está adquiriendo énfasis en los últimos años es la incorporación de estrategias de optimización basadas en métodos de heurística para identificar los mejores hiperparámetros [53]. Dentro de este sentido, los algoritmos como métodos genéticos, simulated annealing o fast greedy suman popularidad, en especial en sus aplicaciones a la identificación de comunidades en estructuras de grafos [35, 43]. No obstante, sus aplicaciones en problemas de ingeniería han sido limitados a casos específicos.

Una vez seleccionados el algoritmo y los hiperparámetros, métricas clásicas de evaluación de desempeño son aplicadas para medir su rendimiento. Se menciona que no existe un consenso sobre cómo medir el desempeño, es decir, si al entrenar un modelo predictivo se divide el conjunto de datos en entrenamiento y validación en razones normalmente de 80:20 o 70:30, no queda claro cuál es el desempeño que se informa. Además, se omite desarrollar estudios estadísticos que permitan dividir n veces el conjunto de datos de entrada, empleando la misma proporción, lo cual, muchas veces, es debido principalmente al costo computacional asociado a dicho proceso. Además, cuando se entrenan los modelos siguiendo una validación cruzada, el rendimiento informado es siempre el promedio. Pero, qué pasa con la desviación estándar?, en qué particiones generadas se logra un mayor desempeño y qué tienen en común dichas particiones? Estas interrogantes son comunes a la hora de desarrollar modelos predictivos. Sin embargo, en muchas ocasiones se dejan de lado debido principalmente al costo que implica responderlas.

Lo expuesto anteriormente, es la base del desarrollo de modelos predictivos empleando un algoritmo de machine learning cualquiera. No obstante, estudios recientes se han enfocado en la aplicación de métodos de ensamble para mejorar el rendimiento de modelos individuales, emulando el comportamiento de los sistemas bagging o boosting, definiéndose formalmente como enssembled learning [46, 144]. No obstante, pese a su utilidad, en ingeniería de proteínas no ha sido tan frecuente su uso, solo existiendo algunos ejemplos en resolución de tareas relacionadas con el estudio de estabilidad térmica de variantes [107, 157]. Interesante mencionar, que los sistemas de ensamble solo se centran en la combinación de modelos predictivos y no exploran también la combinación de variadas representaciones numéricas, lo cual debería poder facilitar el análisis de los modelos desde diferentes puntos de vista.

Con base en estos antecedentes, se diseñó e implementó una estrategia de entrenamiento y desarrollo de modelos predictivos para resolver tareas en ingeniería de proteínas, el cual optimiza la selección de los hiperparámetros y mejora el rendimiento de los modelos combinando tanto las representaciones numéricas desarrolladas en el capítulo anterior, junto con los modelos predictivos de interés. Este enfoque fue evaluado y testeado en diferentes tareas de ingeniería de proteínas y fue comparado con diferentes modelos predictivos previamente desarrollados para dichas tareas, así como también empleando los métodos clásicos de representación y los modelos individuales de predicción entrenados.

Notablemente, el método propuesto sugiere que no solo existe una sinergia entre los codificadores semánticos combinados con representaciones en el espacio de señales, sino que también se crea una sinergia al combinar los diferentes puntos de vista en un modelo predictivo basado en sistemas de ensamble, lo cual demuestra un poder colaborativo entre ellos, incrementando el desempeño de los modelos y favoreciendo su generalización al disminuir las tasas de sobre ajuste.

3.2. Metodología

3.2.1. Descripción del pipeline

La Figura 3.1 muestra el pipeline resumido de la metodología propuesta. De manera general, se tiene que para un conjunto de secuencias inicial, se codifica empleando las representaciones numéricas que se estimen pertinentes, en este caso, se emplearon los codificadores semánticos obtenidos en el capítulo anterior. Una vez las secuencias se encuentran codificadas, técnicas de zero-padding son aplicadas para mantener el mismo largo de vectores. Ya con el conjunto de secuencias codificado, se aplican las transformadas de Fourier para llevar a cabo su digitalización. Luego, diferentes algoritmos de aprendizaje supervisado son explorados siguiendo una estrategia basada en algoritmos genéticos que les permiten seleccionar los hiperparámetros para mejorar su desempeño, generando esto para cada propiedad empleada. Luego, técnicas estadísticas son empleadas para la selección de los modelos predictivos, los cuales son unidos en un único modelo ensamblado, el cual es usado para evaluar su desempeño generando predicción mediante un sistema de votación. De esta forma, se genera el modelo predictivo ensamblado.

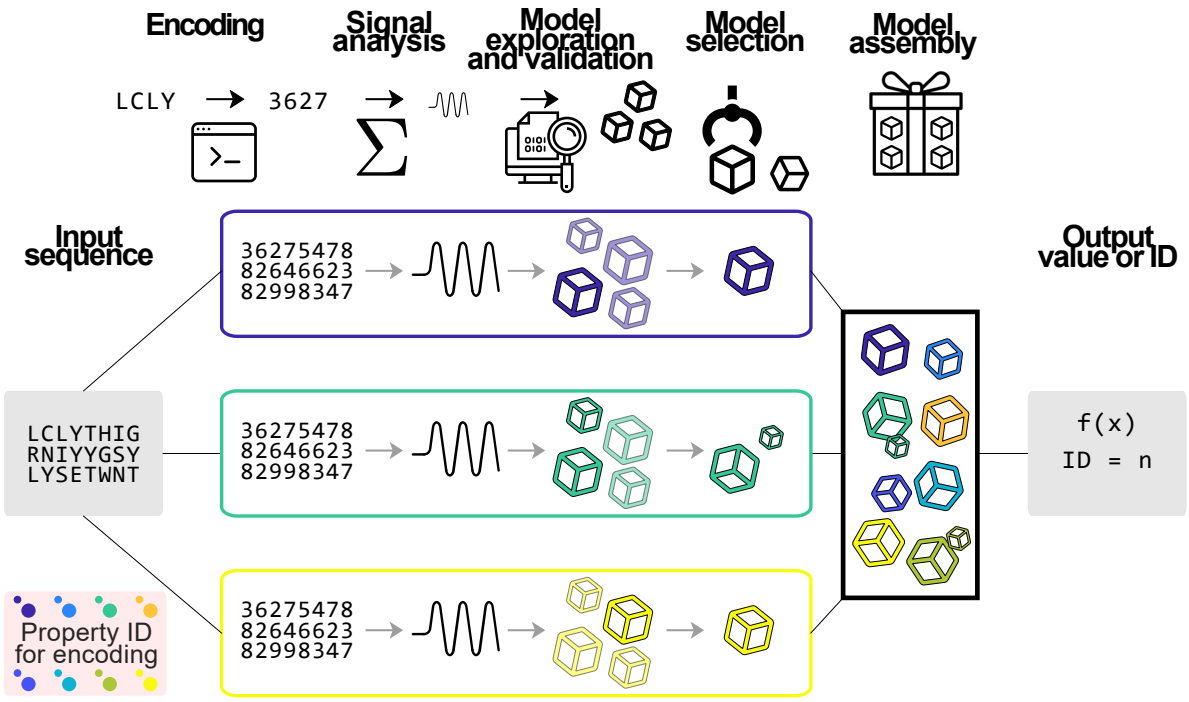


Figura 3.1: Esquema representativo de desarrollo de modelos predictivos ensamblados empleando los codificadores semánticos combinados con transformadas de Fourier.

A continuación, se describen cada una de las etapas más en detalle, con el fin de aclarar ciertos puntos de interés y mencionar los aspectos más relevantes de cada una de ellas.

3.2.2. Codificación y procesamiento de las secuencias

Dado el conjunto de secuencias S recibido como input para el sistema, primero se procesa generando una limpieza de datos, es decir, eliminando las secuencias con residuos no canónicos y aquellas que no presentan respuesta. Luego, se aplica el sistema de codificación de secuencias desarrollado y explicado en el capítulo anterior, en el cual se emplean los codificadores semánticos identificados y se codifican las secuencias. De esta manera, se generan ocho matrices de tamaño $n \times m$ donde n corresponde al número de secuencias y m corresponde al largo de la secuencia. Los valores de m varían de secuencia a secuencia, razón por la cual se aplica una estrategia de *zero-padding* para dejar matrices uniformes en tamaño.

Digitalización

Una vez generada la codificación de secuencias, se someten al proceso de transformación de espacio. Para ello, se aplican las transformadas de Fourier y se transforman las secuencias al espacio de las frecuencias. A partir de esto se genera como resultado una matriz de tamaño $n \times m/2$ donde n es el número de secuencias y $m/2$ el número de puntos en el espectro, esto es debido al funcionamiento de las FFT[1] y además se genera un vector de tamaño $m/2$ el cual corresponde al dominio de las frecuencias. Este proceso se aplica a las 8 matrices generadas en el paso anterior.

Entrenamiento de modelos bajo paradigma de optimización de rendimientos

Una vez generadas las matrices inputs para entrenar los modelos, se someten al proceso de entrenamiento de modelos predictivos con base en algoritmos de aprendizaje supervisado. En esta etapa, diferentes algoritmos de aprendizaje supervisado son explorados, los cuales son resumidos en la Tabla 3.1. Todos los algoritmos son pertenecientes el *machine learning* clásico debido principalmente al volumen de datos generados y a la facilidad de explicar las respuestas que estos tienen.

Algoritmos e hiperparámetros empleados en la etapa de exploración				
#	Algorithm	Type	Parameter	Use
1	Adaboost	Assemble	Algorithm. Number of estimators.	Classification and Prediction.
2	Bagging	Assemble	Bootstrap. Number of estimators.	Classification and Prediction
3	Bernoulli Naive Bayes	Probabilistic	Default.	Classification

[1]Para más detalles, revisar el capítulo 2.

Algoritmos e hiperparámetros empleados en la etapa de exploración				
4	Decision Tree	Characteristics	Division criterion. Impurity Function.	Classification and Prediction.
5	Gaussian Naive Bayes	Probabilistic	Default.	Classification and Prediction.
6	Gradient Tree Boosting	Assemble	Loss Function. Number of estimators.	Classification and Prediction.
7	K-Nearest Neighbors	Distances.	Neighbors Numbers. Algorithm. Metric Distance. Pesos.	Classification and Prediction.
8	ν Support Vector Machine.	Kernel	Kernel. ν. Polynomial degree.	Classification and Prediction.
9	Random Forest.	Assembles	Estimators Number. Impurity Function. Bootstrap.	Classification and Prediction.
10	Support Vector Machine	Kernel	Kernel. C. Polynomial degree.	Classification and Prediction.

Tabla 3.1: Resumen de algoritmos de aprendizaje supervisado explorados y sus exploraciones mínimas.

Cada algoritmo en la Tabla 3.1 se somete a una etapa de exploración de los respectivos hiperparámetros, la cual es guiada por un sistema de optimización de performance, el cual corresponde a un algoritmo genético parametrizado con una inicialización aleatoria, la selección se aplica a los mejores rendimientos según la métrica a optimizar, mientras que como operaciones se trabaja con mutaciones y finalmente el proceso de terminación es hasta que se cumplen los siguientes criterios: i) Se exploraron el máximo de generaciones establecido (1M). ii) La optimización no afecta estadísticamente a los resultados. iii) Se consigue el criterio mínimo de optimización (En este caso, se seleccionó un rendimiento de 95 % y un sobre ajuste entre -0.5 y 0.5). iv) Se alcanza un máximo de memoria.

El proceso considera una etapa de validación cruzada en cada iteración, en la cual se aplica el método de $k - fold$ con un valor dé $k = 10$. Importante mencionar que al entrenar los modelos, se aplica una división aleatoria en una proporción de 80:20, la cual se repite un

total de 100 veces con el fin de dar validez estadística al proceso. De esta forma, una etapa de entrenamiento consiste en una división aleatoria del conjunto de datos en la proporción mencionada, la división de entrenamiento se somete al algoritmo con la estrategia de validación cruzada y la división de validación se ocupa para estimar el rendimiento. Cada etapa mide el rendimiento en ambas secciones (entrenamiento y validación) y se evalúa la tasa de sobre ajuste comparando ambas métricas.

Selección de los modelos

Los modelos entrenados se seleccionan basándose en el rendimiento independiente de cada medida de desempeño. Esto es, para una medida en específico, se toman todos los resultados generados (todas las iteraciones y todos los algoritmos) y se seleccionan los mayores resultados compensando tanto el valor del rendimiento como el sobre ajuste. Una vez seleccionados todos los modelos (algoritmos e hiperparámetros) para cada métrica se agrupan y se les asigna un peso a su predicción con respecto a la cantidad de veces que fue seleccionado, considerando todas las medidas de desempeño utilizadas, esto es, debido a que las métricas representan diferentes formas de evaluar el desempeño, el hecho de ser el mejor en más de una, facilita la incorporación de dichos puntos de vista al sistema de ensamble.

Generación del modelo ensamblado

Por último, todos los modelos predictivos generados son combinados y se desarrolla un único sistema predictivo, este sistema genera sus predicciones con un sistema ponderado, el cual se basa en distribuciones binomiales, con el fin de disminuir el problema de la existencia de desbalance de clases en el caso de existir y con pesos con el fin de favorecer los modelos más predominantes en la selección. Finalmente, el modelo ensamblado se evalúa con un conjunto de datos independiente y se obtiene el rendimiento de la estrategia propuesta.

3.2.3. Casos de uso y conjuntos de datos de prueba

Con el fin de evaluar el rendimiento y la generalización de la metodología propuesta, diferentes casos de estudio y tareas de ingeniería de proteína se ponen a prueba. La Tabla 3.2 describe los conjuntos de datos empleados.

Además, se analizan dos casos de estudio adicionales más en detalle, los cuales se exponen a continuación.

Clasificación de DNA-Binding protein

La clasificación de la proteína de unión al ADN (DBP, por sus siglas en inglés) es uno de los problemas más interesantes de la biotecnología, principalmente debido a sus implicaciones en la ingeniería de proteínas, la biología sintética, la biología molecular y la ingeniería genética

Dataset	Tarea	Tipo de respuesta	Referencia
Enantioselectivity	Predicción	Real	[182]
T50	Predicción	Real	[182]
Solubility	Predicción	Real	[68]
RT prediction	Predicción	Real	[101]
PoP prediction	Predicción	Real	[188]
Anticancer Peptides	Clasificación	Categórica	[183]
Anti Tuberculosis	Clasificación	Categórica	[166]
Binary AMP	Clasificación	Categoríca	[158]
Multi AMP	Clasificación	Múltiple-Categórica	[158]
QSP	Clasificación	Categórica	[132]

Tabla 3.2: Descripción de los sets de datos considerados para la evaluación de la metodología de ensamble propuesta.

[131]. Además, encuentra aplicación directa en la mejora de ADN polimerasas comerciales y enzimas de restricción [170]. Se han propuesto diferentes métodos computacionales para desarrollar modelos de clasificación para la proteína de unión al ADN, que involucran varias estrategias de codificación y caracterización de secuencias. A pesar de los enormes esfuerzos destinados a resolver este problema, permanece abierto. El conjunto de datos para esta tarea se construyó utilizando diferentes conjuntos de datos previamente informados [131, 170, 2], también se eliminaron todas las secuencias sin clasificar, generando un conjunto de datos equilibrado con 1220 ejemplos de proteína de unión al ADN y 1210 proteínas que no se unen al ADN.

Enantioselectividad y termo estabilidad

La enantioselectividad y la termoestabilidad son dos de las propiedades elegidas con más frecuencia para estudiar los efectos de las mutaciones en las proteínas, ya que dependen fuertemente de la secuencia [175]. No obstante, el análisis de los efectos quirales, medidos a partir de la enantioselectividad, es una de las propiedades más complejas de predecir, ya que depende de la secuencia y la preferencia por un tipo de enantiómero. Para verificar la versatilidad y adaptabilidad de la metodología propuesta para predecir variables continuas, se utilizaron los conjuntos de datos de termo estabilidad del citocromo P450 (T50) [94] y de enantioselectividad de la enzima epóxido hidrolasa [182].

3.3. Resultados y discusiones

A continuación, se presentan los principales resultados obtenidos al aplicar la metodología propuesta, así como también las discusiones de cada etapa del proceso y los comentarios generales del sistema predictivo propuesto.

Primero, se compara la metodología propuesta con variadas codificaciones clásicas y con el mejor resultado a la fecha de los reportados para cada tarea de ingeniería de proteínas expuesta en la Tabla 3.2. La Figura 3.2 muestra los resultados obtenidos de esta comparación, exponiendo el coeficiente de Pearson para los modelos de regresión (Figura 3.2 A) y la precisión para los modelos de clasificación (Figura 3.2 B).

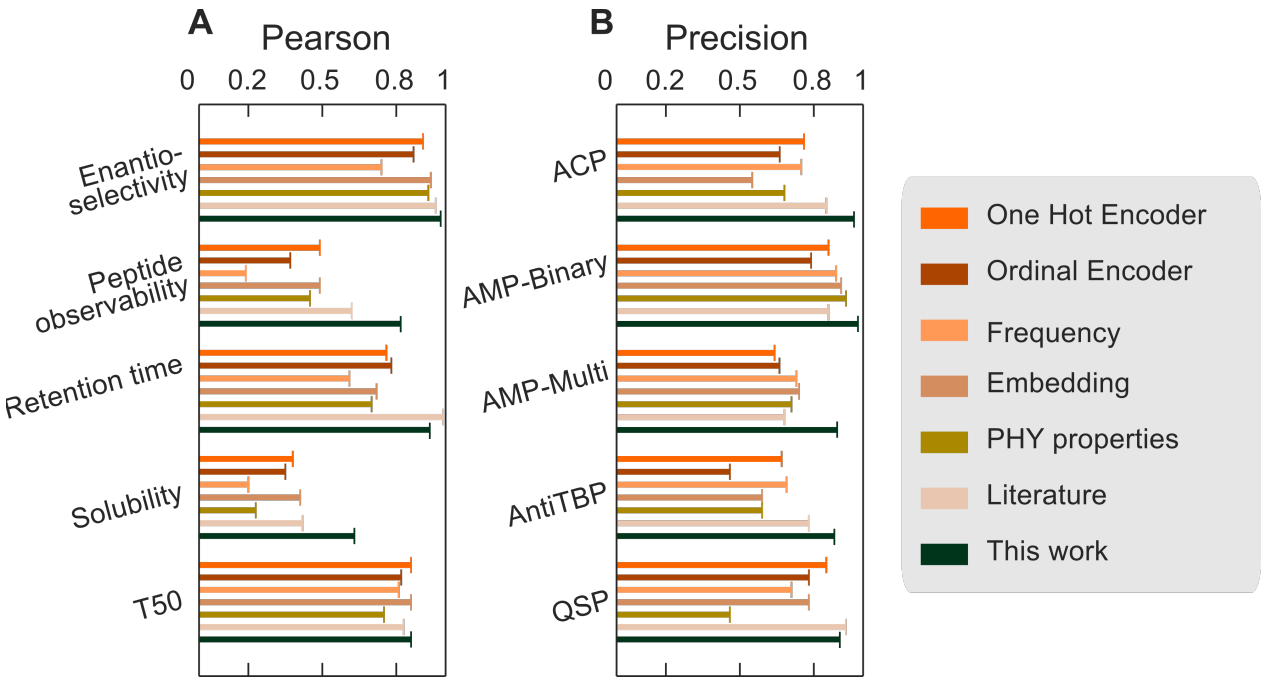


Figura 3.2: Comparación de medidas de desempeño obtenidas mediante la metodología propuesta en este trabajo v/s las diferentes estrategias de codificación exploradas

Tal como se observa en la Figura 3.2, la metodología propuesta presenta mejores resultados en la mayoría de los casos analizados, no solo en comparación contra las codificaciones tradicionales. Si no, que también contra el mejor resultado reportado a la fecha, con excepción de los casos de *retention time* y *quorum sensing peptides (QSP)* en donde los modelos reportados actualmente presentan mejores desempeños de predicción.

Los resultados obtenidos muestran que la metodología propuesta, no solo logra mejor desempeño que los métodos reportados actualmente, lo cual demuestra una alta generalidad, sino que también presenta mejor rendimiento que los métodos de codificaciones actuales.

Con el fin de analizar el aporte del sistema de aprendizaje basado en *ensemble learning*, así como también el efecto de la aplicación de las FFT, se aplica una estrategia similar a los métodos de representación clásica y también se combina con FFT. El resumen de los resultados se observa en la Figura 3.3, en la cual se muestra el coeficiente de Pearson (Figura 3.3 A) y la precisión (Figura 3.3 B), para los modelos de regresión y de clasificación, respectivamente.

La aplicación de FFT sobre las codificaciones de *one hot* y *embedding* disminuyeron su desempeño, tal como se mostró previamente en el capítulo anterior, concordando los resultados con diferentes casos de estudio. No obstante, para el caso de las tareas de predicción del *T50* y la clasificación de *QSP* los resultados mejoraron en comparación a los de la

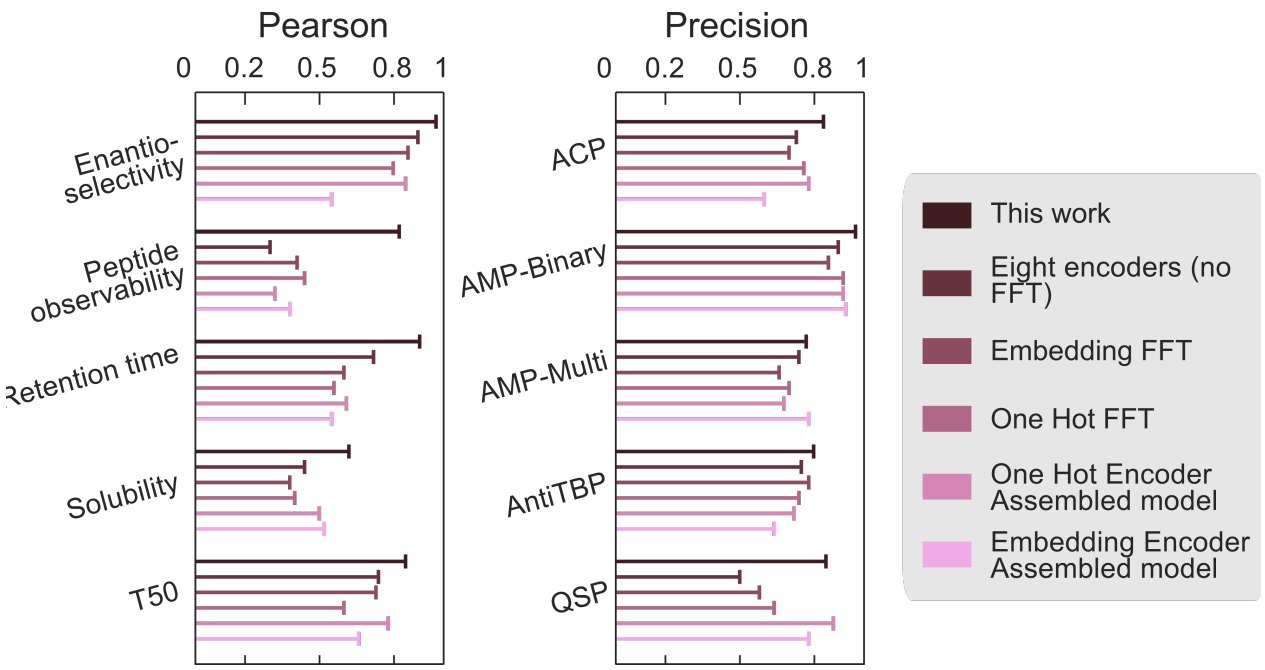


Figura 3.3: Evaluación del efecto de la aplicación de FFT y sistemas de ensamble en los casos de estudio analizados.

etapa inicial, lo cual hace pensar, que en casos particulares el método de ensamble puede ser generalizable para cualquier tipo de representación.

Por otro lado, al analizar los resultados obtenidos aplicando las estrategias de *ensemble learning* se observa que en la mayoría de los casos produce una mejora al rendimiento de modelos individuales, lo cual se observa en ambos tipos de modelos entrenados en la Figura 3.3, comprobando parte de la hipótesis asociada a la aplicación de aprendizaje por ensamble incrementa el desempeño de los modelos predictivos.

Finalmente, se comparó y analizó el efecto individual de cada modelo predictivo contra los resultados obtenido por el modelo generado por aprendizaje ensamblado. Para ello, se diseñó e implementó un sistema de clasificación de proteínas de unión a DNA, cuyos resultados se observan en la Figura 3.4.

Tal como se observa en la Figura 3.4, el modelo ensamblado presenta mejores resultados de desempeño en comparación a las propiedades individuales, lo cual se aprecia tanto en los análisis de sensibilidad (Figura 3.4 A) como de especificidad (Figura 3.4 B). Esto puede deberse a que se genera una sinergia entre los diferentes puntos de vista empleados para representar/caracterizar las secuencias, lo cual al desarrollar modelos predictivos bajo técnicas de ensamble, se ven favorecidas y aumentan el rendimiento de los sistemas de predicción. Esto se aprecia también en los casos analizados previamente, lo cual demuestra la existencia de una sinergia, provocando una *"ganancia de información"* a la hora de emplear los diferentes codificadores semánticos generados.

De esta forma, se implementa una metodología de modelos ensamblados, los cuales se favorecen al usar diferentes puntos de vista empleando los codificadores semánticos. Sin embargo, el hecho de aumentar el desempeño, dado a la optimización con base en los sistemas

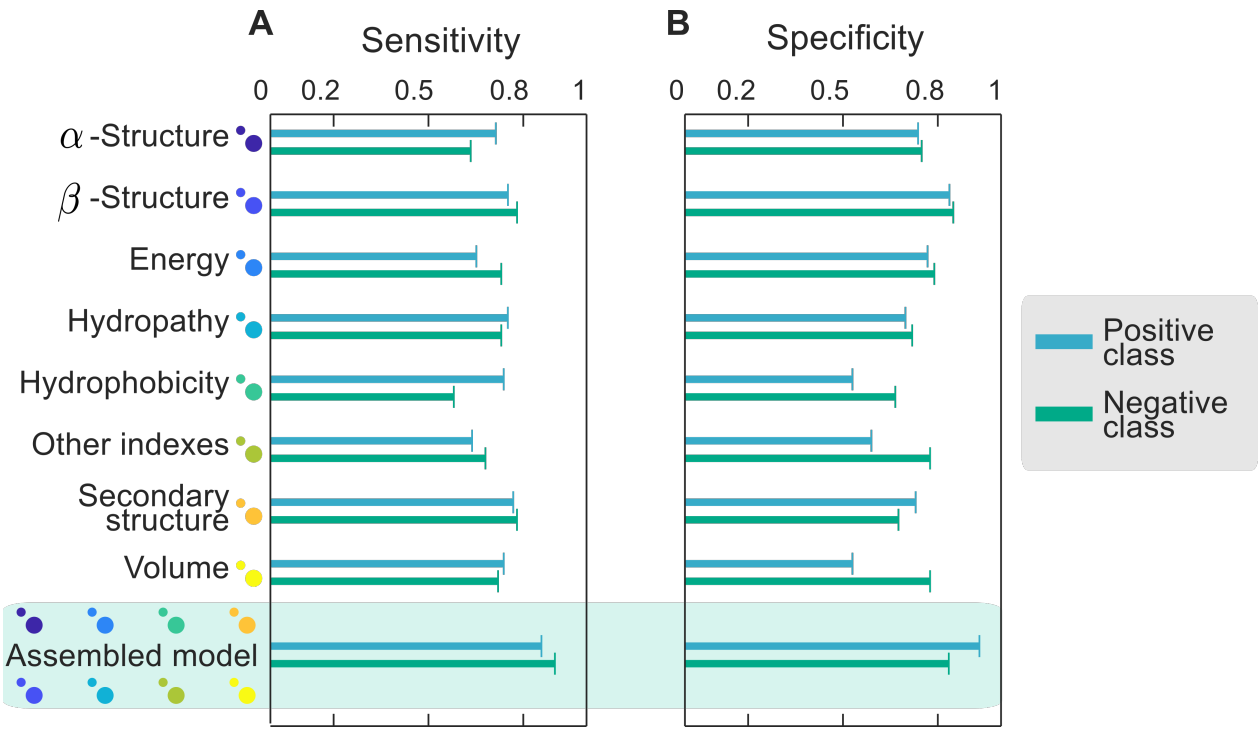


Figura 3.4: Análisis del efecto individual versus el modelo ensamblado sobre la sensibilidad y la especificidad.

heurísticos de algoritmos genéticos propuestos, puede provocar sobre ajuste al adaptar de sobremanera los hiperparámetros de configuración al sistema de datos inicial. El análisis de este efecto se evalúa con respecto al estudio de enantioselectividad y estabilidad térmica de los conjuntos de datos propuestos por [182] que corresponden a librerías de evolución dirigida, donde las secuencias corresponden a variantes mutacionales con un máximo de 5 residuos mutados al mismo tiempo.

La Figura 3.5 muestra los resultados obtenidos para el análisis de sobre ajuste. En la Figura 3.5 A se observan los *scatter plot* de precisión v/s realidad para los modelos ensamblados generados. Se puedo determinar visualmente que hay una tendencia de los resultados a generar predicciones inferiores a la realidad para el caso de modelos de estabilidad térmica, lo cual también ocurre con los modelos de enantioselectividad. Dado esto, se propone una curva de calibración para las predicciones y se mide nuevamente el desempeño, en ambos casos bajan las métricas a un **R-Score** de **0.86** y de **0.89**, respectivamente. Las curvas muestran una tendencia más homogénea y no una tendencia clara de predicción, disminuyendo el sobre ajuste identificado.

Es importante mencionar que pese a que los métodos desarrollados tratan de evitar la existencia de sobre ajuste o disminuir su existencia, cuando la cantidad de ejemplos son insuficientes o la información no es lo suficientemente representativa, es inevitable caer en este tipo de problemáticas. El desarrollo de las metodologías planteadas, se propuso para estudio de proteínas, no de mutaciones puntuales en sí, debido a que depende de una variabilidad mínima en el conjunto de datos. De esta forma, si el conjunto de datos pertenece a librerías de mutaciones puntuales, tales como el caso de las tareas de predicción de estabilidad térmica y de efectos sobre la enantioselectividad, existirá una tendencia al sobre ajuste. Esto también

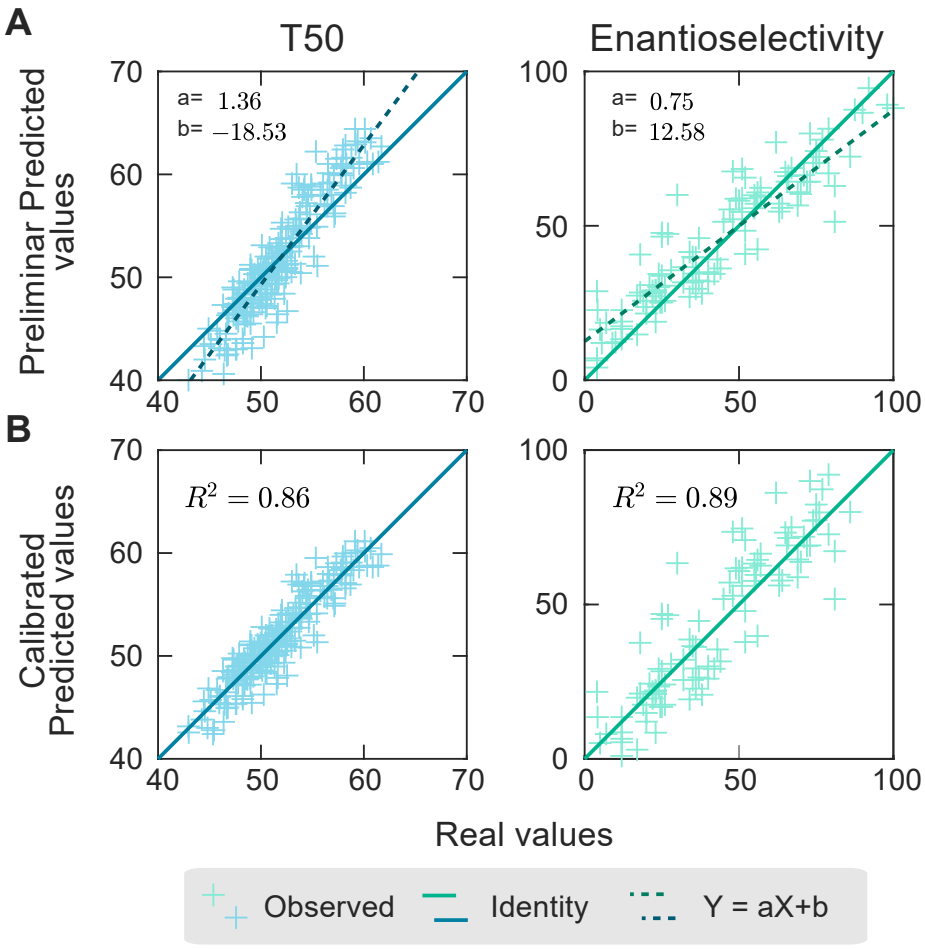


Figura 3.5: Análisis del efecto individual versus el modelo ensamblado sobre la sensibilidad y la especificidad.

puede darse en los casos de conjuntos de datos basados en *deep mutational scanning* o librerías de evolución dirigida. Este tipo de problemáticas será abordado en los siguientes capítulos de este trabajo de tesis.

3.4. Conclusiones y principales comentarios

Una vez finalizada la metodología propuesta y validada desde diferentes aristas, se pueden concluir los siguientes puntos.

1. Se diseñó e implementó una metodología de desarrollo de modelos predictivos basada en sistemas de aprendizaje por ensamble, la cual combina modelos de optimización para la selección de hiperparámetros, estadística para el desarrollo de clasificaciones y los codificadores semánticos representados desde el punto de vista de espacio de señales.
2. La metodología propuesta se valida y compara en diferentes tareas de ingeniería de proteínas, tanto en tareas de regresión como de clasificación, logrando mejores resulta-

dos en la mayoría de los casos, no solo las codificaciones clásicas, sino que también en mejora el rendimiento con respecto a los modelos actuales específicos reportados en el estado de arte, demostrando ser una metodología potente y generalizable, aplicable a diferentes sistemas de interés.

3. Los métodos de ensamble favorecen el incremento de desempeño de modelos predictivos, independiente de su representación. Sin embargo, las FFT al aplicarlas sobre representaciones clásicas no produce resultados correctos. Demostrando que, el punto de vista de interés se encuentra no solo en la combinación de modelos, sino que también en la combinación de propiedades.

4. Se demostró la sinergia entre la combinación de los diferentes puntos de vista (codificadores semánticos combinados con FFT y el entrenamiento de modelos ensamblados), mostrando que individualmente los modelos generados con una propiedad en particular logran un menor desempeño que combinados, tanto en sensibilidad como en especificidad.

5. Al evaluar la metodología propuesta en conjuntos de datos de mutaciones provenientes desde evolución dirigida o técnicas de *screening* similares, se identifica que si bien se logra un buen desempeño en comparación al estado de arte, se devalúa su valor basándose en el sobre ajuste existente en las predicciones, lo cual se debe principalmente a la poca información y cantidad de registros con los que se evaluó los problemas asociados a la estabilidad térmica y enantioselectividad, desarrollando sistemas de calibración para disminuir el sobre ajuste. Pese a esto, es una técnica robusta y los resultados son altamente competitivos con las técnicas más recientes de desarrollo de modelos predictivos, siendo una estrategia válida para incorporarla a sistemas de diseño de variantes.

Capítulo 4

Estrategias de diseño de secuencias con propiedades deseables

El diseño de secuencias amino acídicas con propiedades deseables es una de las metas más relevantes de la ingeniería de proteínas y a su vez uno de los grandes desafíos de las últimas cuatro décadas. Las técnicas de evolución dirigida y diseño racional han sido los pilares para resolver este desafío. Sin embargo, presentan problemas relevantes a la usabilidad, nivel de conocimiento y generalización, lo cual resulta complejo de aplicar en variados casos [157]. Con la llegada del *machine learning*, ambas técnicas se han visto beneficiadas en sus protocolos debido a la incorporación de estas estrategias, naciendo metodologías como el *machine learning directed evolution* [174] y el diseño semi-racional [153].

Actualmente, existen diferentes enfoques para diseñar secuencias o variantes *in-silico*, centrándose principalmente en estudio de variantes mutacionales mediante estrategias de bioinformática estructural, empleando herramientas como *Fold-X* [148] o SDM [121], mientras que desde el punto de vista de ML se han destacado los esfuerzos por incorporar los sistemas predictivos y guiar las técnicas de diseño de proteínas [174]. Recientemente, las estrategias basadas en *deep generative models* han sido empleadas para generar secuencias desde el contexto de aprendizaje [176], demostrando su usabilidad en diferentes tareas de ingeniería de proteínas, siendo ejemplos relevantes el desarrollo de secuencias de péptidos señal [176], así como también la elaboración de mutaciones en la proteína P53 [97]. No obstante, las aplicaciones de estos métodos requieren grandes volúmenes de información para poder generalizar los comportamientos y desarrollar las secuencias bajo el contexto deseado.

Otra forma de diseño y reconocimiento de variantes prometedoras se ha basado en la exploración de *landscapes*, es decir, simular todas las posibles mutaciones puntuales o dobles de una proteína de interés. No obstante, evaluar combinatorias de este tipo es una tarea ardua y compleja computacionalmente. Métodos como *EcNet* [97] y *UniRep* [166] se han centrado en la elaboración de estrategias que faciliten la reconstrucción de los *landscape* desde un punto de vista enfocado en las técnicas de *deep learning* y el desarrollo de diccionarios adquiridos mediante aprendizaje, empleando técnicas de lenguaje natural, definiendo un *lenguaje de proteínas*. Por otro lado, los métodos habilitados como *frameworks* de implementación de modelos tales como los propuestos por [157] y [107] también han sido empleados para explora-

ción de *landscape*, permitiendo simular eficientemente mutaciones para análisis de estabilidad térmica. Sin embargo, los estudios para tareas más complejas son difíciles de abordar debido principalmente a que las relaciones no son lineales y los patrones de sitios relevantes para mutar no son claros. Además, solo los métodos basados en *deep generative models* han sido habilitados para generar secuencias *de novo*.

Con la llegada de *AlphaFold* y *RoseTTAfold* [11] es posible evaluar estructuralmente los modelos generados. Sin embargo, su aplicación es costosa computacionalmente y a la fecha, no está habilitada para el trabajo con multímeros o secuencias más complejas. No obstante, contar con herramientas de validación estructural brinda un soporte valioso a los sistemas de diseño. Sin embargo, pese a los grandes avances tanto a nivel bioinformático como de *machine learning* en la ingeniería de proteínas, los sistemas de diseño de secuencias no han sido explotados, solo centrándose en la exploración de *landscapes* y siendo empleados en tareas bien establecidas de la ingeniería de proteínas.

Con base en lo anterior, el tópico de este capítulo se centra en la presentación de una estrategia computacional basada en la exploración de espacios latentes, la cual puede usarse para reconstrucción de *landscapes*, exploración de secuencias mediante técnicas estadísticas y evaluación de las propiedades deseables mediante predicción con sistemas híbridos de ensamble combinados con codificadores semánticos y representaciones de Fourier, centrándose principalmente en dichas representaciones para formar los modelos estadísticos de pertenencia al espacio latente de la propiedad de interés.

La metodología propuesta fue empleada en diferentes aplicaciones, contemplando i) Reconstrucción de fitness de termo estabilidad de proteínas. ii) Diseño de secuencias de péptidos con actividad biológica deseable. iii) Diseño de péptidos con actividad anti-VIH para incremento de actividad farmacológica. iv) Diseño de péptidos con productividad deseable bajo sistema de producción recombinante.

Finalmente, si bien se aprecian diferentes puntos de casos de estudio, existe una metodología base, la cual se fue adaptando o incorporando nuevas reglas, con base en la cantidad de registros o el tipo de desarrollo a analizar.

De esta forma, se propone una metodología que combina tanto las representaciones de espacios de señales con base en codificaciones semánticas (expuestas en el capítulo 2), el diseño e implementación de modelos predictivos ensamblados (expuestos en el capítulo 3) y las estrategias de análisis de espacios latentes y métodos probabilísticos generados en este capítulo.

4.1. Metodología

Con el fin de llevar a cabo los diferentes objetivos planteados, se resume la metodología propuesta, la cual será dividida para las diferentes fuentes de interés, contemplando desde la exploración y reconstrucción de *landscapes*, hasta la elaboración de sistemas estadísticos de predicción de secuencias para incorporarlas a sistemas experimentales, cada una de ellas, se expone a continuación.

4.1.1. Diseño e implementación de herramientas de exploración de landscapes

Los *landscapes* corresponden a posibles cambios en la secuencia a nivel *single point mutation* o dos o más mutaciones al mismo tiempo [107], los cuales se asocian a evaluar el efecto que provoca la sustitución de él o los aminoácidos en la secuencia *wild type* con respecto al tipo de respuesta o propiedad que se desea analizar. Para este caso, se evaluaron cambios en los efectos termodinámicos de la proteína *dihydrofolate reductase*. Para ello, primero se recopilan diferentes mutaciones reportadas de esta proteína desde variadas fuentes de información [18, 126, 9, 20]. En una segunda instancia, se implementan modelos predictivos ensamblados siguiendo las estrategias propuestas en el capítulo anterior. Finalmente, se exploran *landscapes*, simulando mutaciones puntuales en la proteína de interés y se evalúan clasificándolos con respecto a categorías definidas, en este caso, si aumenta la estabilidad, es neutra o disminuye la estabilidad, esto es con respecto al valor de $\Delta\Delta G$ empleado para cuantificar la estabilidad térmica de la proteína.

4.1.2. Diseño e implementación de estrategias de exploración de secuencias de péptidos para evaluar actividades biológicas deseables

Los péptidos antimicrobianos (AMP) son conocidos como péptidos de defensa del huésped [158]. Estas moléculas juegan un papel fundamental en la respuesta inmune innata, por lo que tienen aplicación directa en las áreas farmacéutica, biotecnológica e industrial [123, 101]. Se han desarrollado diferentes métodos computacionales basados en ML para clasificar los péptidos antimicrobianos [178, 32, 183, 188]. En este caso de estudio, se usó las secuencias peptídicas reportadas en *PeptipediaDB* [130] para desarrollar modelos de clasificación de péptidos AMP.

Una vez implementado los modelos predictivos, se diseña un modelo estadístico de evaluación de *landscape*, el cual se basa en los siguientes puntos.

- Sea P el conjunto de proteínas a evaluar y n el número de propiedades semánticas a emplear.
- Para cada propiedad $P_i \in P$ se puede obtener los espectros de frecuencia estadísticos con base en las transformadas de Fourier de las secuencias. Para ello, se estima en cada punto del espectro los valores promedios e intervalos de confianza al 95 %. De esta forma, se construye un ancho de banda de espectro de frecuencias para estudiar la propiedad P_i.
- Una vez construidos todos los anchos de banda estadísticos, se implementa un sistema de evaluación de probabilidades de pertenecer al ancho de banda de una propiedad en específica. Esto es, para una nueva secuencia, se estima la probabilidad de existir dentro del ancho de banda de una propiedad P_i, contando el número de veces que se encuentra dentro y dividiéndolo por el número total de puntos.

- Una vez se tiene estimada la probabilidad para una secuencia para todas las propiedades analizadas o empleadas, se estima la probabilidad conjunta de una secuencia de pertenecer a la combinación de los anchos de banda de las propiedades empleadas. Para ello, se hace la multiplicación de las probabilidades para obtener esta probabilidad conjunta debido principalmente a que las propiedades seleccionadas desde los componentes semánticos expuestos en el capítulo 2 resultan ser independientes entre sí, lo cual facilita este procedimiento.
- Finalmente, se evalúa cada secuencia con base en un umbral de clasificación, el cual se puede variar a medida que se vaya explorando y mejorando la estrategia para casos definidos.

Una vez desarrollados los modelos y definidas las reglas de evaluación, se seleccionan al azar secuencias desde la base de datos *PeptipediaDB* [130] y se utilizaron para explorar actividades biológicas, con el fin de validar la metodología propuesta.

4.1.3. Diseñando modelos en conjuntos de secuencias poco informativos y explorando sistemas probabilísticos de predicción

Contemplando como inputs conjuntos de secuencias generadas desde el sistema de producción de péptidos *MachinePep*[1], así como también péptidos con actividad inhibitoria de VIH (virus de la inmunodeficiencia humana) obtenidos desde la literatura y bases de datos previamente reportadas y recopiladas en *PeptipediaDB* [130], se diseñó e implementó una estrategia de entrenamiento de modelos predictivos y análisis de espacio latente con el fin de generar secuencias de péptidos con propiedades deseables según el caso de interés, es decir, ya sea, para evaluar péptidos en el sistema de producción *MahcinePep* o para diseñar nuevas secuencias de péptidos con actividad inhibitoria de VIH. Para llevar a cabo la metodología planteada se elaboraron y validaron variadas estrategias computacionales soportadas por métodos de *machine learning*, técnicas estadísticas y métodos de representaciones numéricas de secuencias de proteínas basados en propiedades fisicoquímicas combinadas con procesamiento digital de señales. Por otro lado, la falta de información y escasez de datos de los conjuntos de entrada requirió la elaboración de estrategias de *over sampling*, aplicando variaciones de *data augmentation*.

A continuación, se describe brevemente la metodología generada para cada etapa, generando divisiones de cada problemática cuando corresponde.

Conjunto de datos y los principales problemas a tratar

Dependiendo del problema a trabajar, son diferentes los conjuntos de datos iniciales con los que se contaba. A continuación, se detallan brevemente las principales características de cada uno de estos.

[1]Sistema desarrollado en CeBiB para el cual se colabora desarrollando métodos computacionales

Conjunto de datos para evaluar productividad en sistema de producción MachinePep

El conjunto de datos inicial a trabajar posee un total de 41 secuencias de péptidos en lenguaje amino acídico, con un número de residuos entre 11 y 72. La variable a predecir es la productividad, la cual se expresa en mg/L cuyos valores varían entre 0.56 y 36.33, con un promedio de 8.07. Todos estos péptidos fueron obtenidos a partir del sistema de producción recombinante *MachinePep*. Existen dos problemas complejos a la hora de trabajar con este tipo de conjuntos de datos. El primero se basa en las estrategias de entrenamiento a desarrollar, mientras que el segundo es con respecto a la baja cantidad de ejemplos existentes, ya que, con 41 ejemplos, generar modelos predictivos con alta generalidad es bastante complejo. No obstante, técnicas de *over sampling* pueden ser aplicadas con el fin de incrementar la cantidad de ejemplos.

Conjunto de datos para evaluar eficiencia de actividad inhibitoria en péptidos VIH

En el caso del problema de diseño de péptidos con actividad VIH, 980 secuencias de péptidos con esta actividad biológica fueron recolectadas desde diferentes fuentes de información, en especial desde la base de datos de péptidos *PeptipediaDB* [130]. Todas estas secuencias contaban con un valor de actividad asociado a IC50. No obstante, existen secuencias con un valor categórico de dicha instancia, las cuales fueron descartadas, trabajando con un total de 458 secuencias. Importante destacar, que los valores de medición de la actividad farmacológica fueron escalados para tenerlo en las mismas unidades de medida (nM). Las secuencias varían en un largo entre 5 y 150 residuos. Además, como condición especial, todos los péptidos identificados deben tener actividad biológica antimicrobiana y antiviral, esto es basándose en el árbol de categorización propuesto en *PeptipediaDB* [130].

Preparación del conjunto de datos

A continuación, se describe brevemente cuáles fueron las estrategias desarrolladas para preparar los conjuntos de datos para el entrenamiento de los modelos predictivos, estos se describen por separado con el fin de facilitar la comprensión de los puntos relevantes en cada uno de los casos particulares.

Preparación de conjunto de datos para estimación de productividad

En este caso, en una primera instancia se binariza las respuestas de interés (la productividad) con el fin de generar categorías. Este procedimiento se llevó a cabo considerando los valores estadísticos de la muestra de interés. Generando la discretización con base en:

- Si el valor es mayor al promedio, entonces es clase 1.
- En caso contrario es clase 0.

El fundamento tras esta decisión fue guiado por los especialistas que desarrollaron la metodología de producción, en este caso, el grupo de biología molecular. La idea general de esto era generar conjuntos de datos equilibrados sin desbalance de clases que pudieran afectar negativamente a los modelos a generar.

Preparación de conjunto de datos para estimación de actividad inhibitoria de péptidos VIH

En el caso del conjunto de datos de secuencias con actividad inhibitoria para VIH. Primero, se eliminaron las secuencias con respuesta categórica o medición cuantitativa de la actividad inhibitoria para VIH, esto es, todos aquellos que no tuviera valor numérico como respuesta. Además, se eliminaron registros que presentaban valores asociados a cuantificación de pérdida de plegamiento o estabilidad de la proteína GP41 o del complejo de interacción que se forma, debido principalmente a que no se contaba con información suficiente para escalar dichos valores y llevarlos a un valor cuantitativo en unidades de medidas deseables.

Luego, todos los valores se escalaron a una única medida y se generaron categorías para facilitar el entrenamiento de modelos predictivos asociados al desarrollo de sistemas de clasificación. Esta categorización se basó en los siguientes puntos con respecto al IC50 (medida farmacológica empleada).

- Si el valor es menor al cuartil 1 (25 %) es categorizado como C1.
- Si el valor es menor al cuartil 2 (50 %) y mayor o igual al cuartil 1 se categoriza como C2.
- Si el valor está entre el cuartil 2 y el cuartil 3 (75 %) se categoriza como C3.
- En caso contrario se categoriza como C4.

Estas categorías representan niveles dentro de la distribución, siendo el nivel más importante las C1, ya que representan los mejores valores para actividad VIH. Se selecciona la distribución de cuartiles para generar la categorización, debido principalmente a la generación de conjuntos de datos equilibrados.

Representación numérica del conjunto de datos y estrategias de data augmentation

Una vez se tiene el conjunto de datos discretizado (con base en las categorías de clasificación generadas en cada caso expuesto) comienza el proceso de la representación numérica de las secuencias, lo cual es requerido para trabajar con modelos predictivos. Tal como se apreció en el capítulo dos, existen diferentes estrategias de codificación de secuencias. Además, acoplado al problema de representación, existe el problema de la cantidad insuficiente de datos para entrenamiento de modelos predictivos, en particular, para el caso de los modelos de predicción de productividad para el sistema de producción *MachinePep*. Con base en lo anterior, se propone una estrategia de representación numérica, la cual involucra las propiedades

fisicoquímicas seleccionadas mediante técnicas de *natural language processing*[2] y métodos de *data augmentation* para el caso del conjunto de datos con menor cantidad de información.

Un resumen de la metodología planteada se expone en la Figura 4.1, la cual es descrita a continuación.

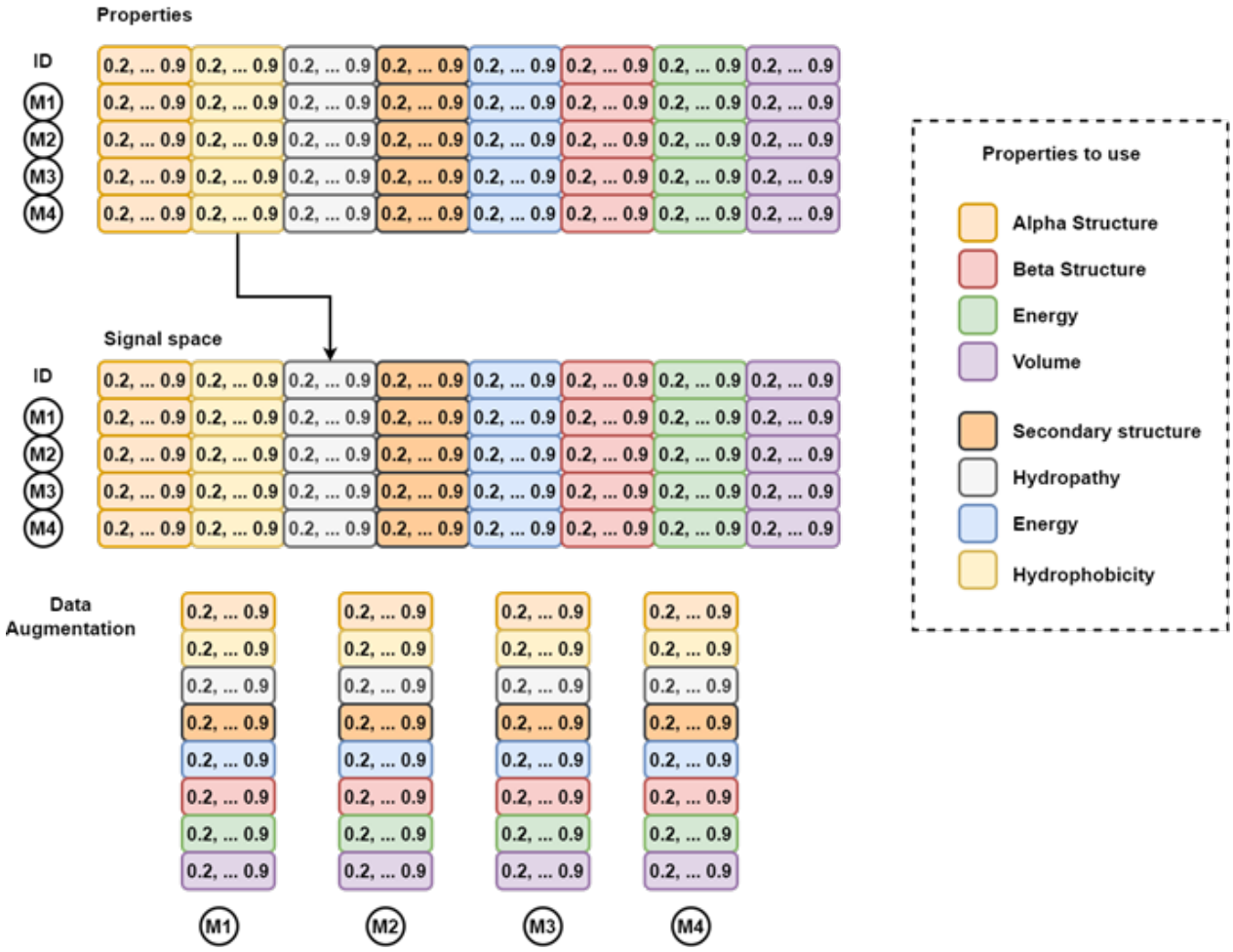


Figura 4.1: Descripción del proceso de representación numérica

Primero, empleando las codificaciones provenientes de propiedades fisicoquímicas propuestas en el capítulo dos de este trabajo de tesis, se codifican las secuencias obteniendo 8 vectores independientes para cada secuencia en el conjunto de datos. Estos vectores tienen la misma cantidad de elementos como residuos tiene la secuencia a codificar. La codificación contempla de que cada aminoácido R en la secuencia S se codifica por un valor $P(R)$ donde P representa a los codificadores propuestos para una propiedad en particular. De esta forma, cada residuo tiene un único valor dependiendo de la propiedad. Importante destacar, es que se requiere la aplicación de técnicas de *zero-padding* para poder ajustar los largos de los vectores codificados y estandarizarlos en uno único, lo cual es requisito para aplicar algoritmos de *machine learning*. Luego, cada vector es transformado aplicando métodos de digitalización, con el fin de llevarlo a representaciones del espacio de señales, empleando transformadas, según lo descrito en el capítulo dos de este trabajo de tesis. Una de las ventajas de esta representación

[2]Para más detalles revisar el capítulo dos de este trabajo de tesis

es que permite emular los comportamientos estructurales y las interacciones de los residuos dado el enfoque del espacio de frecuencias, en el cual, cada punto está relacionado con el resto de la distribución, lo cual en proteínas es aludido a que cada residuo tiene una relevancia sobre otro.

Finalmente, y solo para el caso del entrenamiento de modelos predictivos de productividad en el sistema *MachinePep*, una vez generado los vectores, estos se acoplan en una única matriz y se genera un único conjunto de datos, a diferencia del caso del desarrollo de sistemas de predicción para actividad inhibitoria de VIH, donde se desarrollan 8 conjuntos independientes.

Entrenamiento de modelos predictivos y desarrollo de sistemas ensamblados

El entrenamiento de modelos predictivos para cada caso, se basó en la metodología de ensamble propuesta en el capítulo tres de este trabajo de tesis. Es decir, se exploran diferentes algoritmos e hiperparámetros guiados por un sistema de optimización heurística, los cuales se combinan en un único sistema predictivo con base en métodos de aprendizaje por ensamble.

De esta forma se genera un modelo predictivo ensamblado. Se menciona que, dos etapas se modifican al pipeline propuesto de entrenamiento de modelos predictivos ensamblados.

1. **División del conjunto de datos en entrenamiento y en validación**. Normalmente, se divide el conjunto de datos input en dos, uno empleado para entrenar el modelo y otro para ocupar el modelo, obtener las predicciones y estimar el rendimiento. Normalmente, esta división se hace en proporciones 70:30 u 80:20. El conjunto de datos empleado para entrenamiento se somete a una validación cruzada con $K = 10$ para poder detectar sobre ajuste. En este sentido, las diferencias aplicadas en este caso, en específico, se centran en dividir el conjunto inicial en una proporción de **90:10** y aplicar una validación cruzada **leave one out** para la etapa de entrenamiento.
2. **División iterativa del conjunto input**. Tal como se nombró en el punto anterior, se hace una división de 90:10 en conjunto de entrenamiento y validación. Este proceso se itera en $n = 100$ instancias para brindar soporte estadístico al proceso desarrollado.

Hasta este punto, ambos desarrollos de modelos predictivos son iguales. Sin embargo, debido a las diferentes estrategias de desarrollo de los conjuntos de datos, divergen en este punto. A continuación, se explica la continuación del proceso separadamente para cada desarrollo.

Desarrollo de modelos predictivos con data augmentation

Para el caso de los modelos predictivos de clasificación de productividad en el sistema de producción *MachinePep*. Se emplean los modelos seleccionados con mayor rendimiento y menor sobre ajuste, para someterlos a un nuevo proceso de entrenamiento. Sin embargo, este

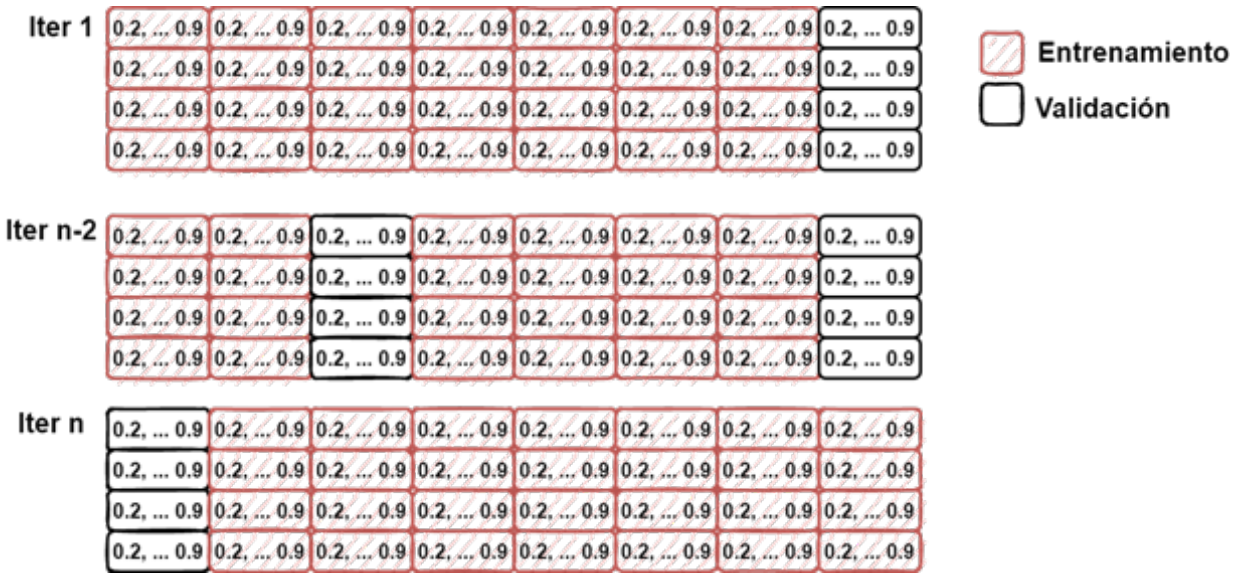


Figura 4.2: *Leave one peptide out* como estrategia de entrenamiento de modelos predictivos

entrenamiento se basa en una metodología propuesta para este proyecto denominada *Leave One peptide Out*, la cual se detalla esquemáticamente en la Figura 4.2.

Este proceso se puede comentar los siguientes puntos de interés.

- En el conjunto de datos, cada secuencia de péptidos está sobre expresado, es decir, 8 ejemplos del conjunto representan un único péptido, esto es debido al *data augmentation* generado. Los rendimientos obtenidos se basan principalmente en una representación, es decir, que los rendimientos pueden estar sesgados dado que propiedades pueden ser más informativas en algunos casos, es decir, representaciones específicas de las secuencias de péptidos pueden ser más informativas.
- Dado esto se vuelve a entrenar el conjunto de datos, pero aplicando la validación propuesta, para ello, se contemplan los siguientes puntos de interés.
  - En cada iteración se deja fuera un péptido completo, es decir, las 8 representaciones de un péptido en específico se dejan fuera de la etapa de entrenamiento del modelo.
  - Luego, se entrena el modelo según la configuración de algoritmo e hiperparámetros que corresponda dado a la selección desarrollada. Este proceso se entrena con validación cruzada, manteniendo la lógica expuesta en la primera etapa.
  - Luego, empleando el modelo generado, se obtiene las predicciones para el péptido dejado fuera, es decir, para las 8 representaciones. Estas predicciones se comparan con los valores reales y se obtiene el rendimiento (Esto es solo de validación, dado que la diferencia debería ser mínima). Además, se obtiene la clasificación final del péptido por un sistema de votación. Es decir.
    * Tomando en consideración cada predicción se hace una ponderación en modo votación y se obtiene la clasificación final del péptido haciendo dicha votación (Simplemente el que tiene más votos, en caso de empate es un *random*).

        * Se compara la clasificación por ponderación con la clasificación real y se obtiene almacenan los valores
    - Una vez iterado para cada péptido se emplean los valores almacenados y se obtiene el rendimiento final del proceso.

- Finalmente, este proceso se desarrolla para todos los modelos seleccionados en la etapa anterior. A partir de lo cual, una vez teniendo el desempeño, se selecciona el mejor modelo simplemente por selección de la medida de desempeño más alta.

Desarrollo de modelos ensamblados para péptidos con actividad inhibitoria VIH

Para el caso de los sistemas predictivos se siguió la misma metodología de entrenamiento de modelos ensamblados expuesta en el capítulo tres, no registrando cambios significativos en el pipeline propuesto. No obstante, como consideración especial solo se emplearon algoritmos basados en técnicas de *bagging* o *boosting*, así como también, algoritmos de árboles de decisión, ignorando los métodos basados en kernel o distancias (SVM y KNN), debido a que se deseaba comprender la relación entre las propiedades, los puntos de interés y la identificación en las reglas asociadas a las decisiones de clasificación para las secuencias de interés.

Desarrollo de espacios latentes e implementación de modelos probabilísticos

Finalmente, con la metodología de representación numérica y los modelos predictivos entrenados, es posible desarrollar estrategias de exploración de espacios latentes para evaluar nuevas secuencias. La Figura 4.3 muestra un esquema representativo de la propuesta generada. Esta metodología se puede dividir en cuatro grandes etapas, las cuales se explican a continuación.

Los puntos 1 y 2 se centran en obtener un espacio de distribución. Esto hace referencia a generar una distribución de probabilidad por cada punto existente en el espacio de codificaciones generado. Para obtener este espacio se contempla lo siguiente.

- Obtener un intervalo de confianza por cada punto en una forma de representación (propiedad fisicoquímica)
- Armar un ancho de banda para cada propiedad empleando los diferentes intervalos de confianza generados.
- Generar anchos de banda por cada propiedad utilizada

En los puntos tres y cuatro se contemplan la evaluación de nuevos conjuntos de secuencias, los cuales se deben codificar empleando las estrategias nombradas y evaluar en los anchos de banda. Para ello se debe.

- Por cada propiedad, estimar la probabilidad de pertenecer al ancho de banda de la propiedad correspondiente. Para esto, se estima la cantidad de veces que un punto se encuentra dentro del ancho de banda y se divide por la cantidad de puntos totales.

- Evaluar la probabilidad total, contemplando para ella cada probabilidad de propiedad independiente y haciendo la multiplicación de las ocho propiedades. De esta forma, se tiene una probabilidad única.
- Finalmente, si la probabilidad supera cierto umbral, se aplica el modelo predictivo (ensamblado o por *data augmentation*) y se obtiene la respuesta de interés.

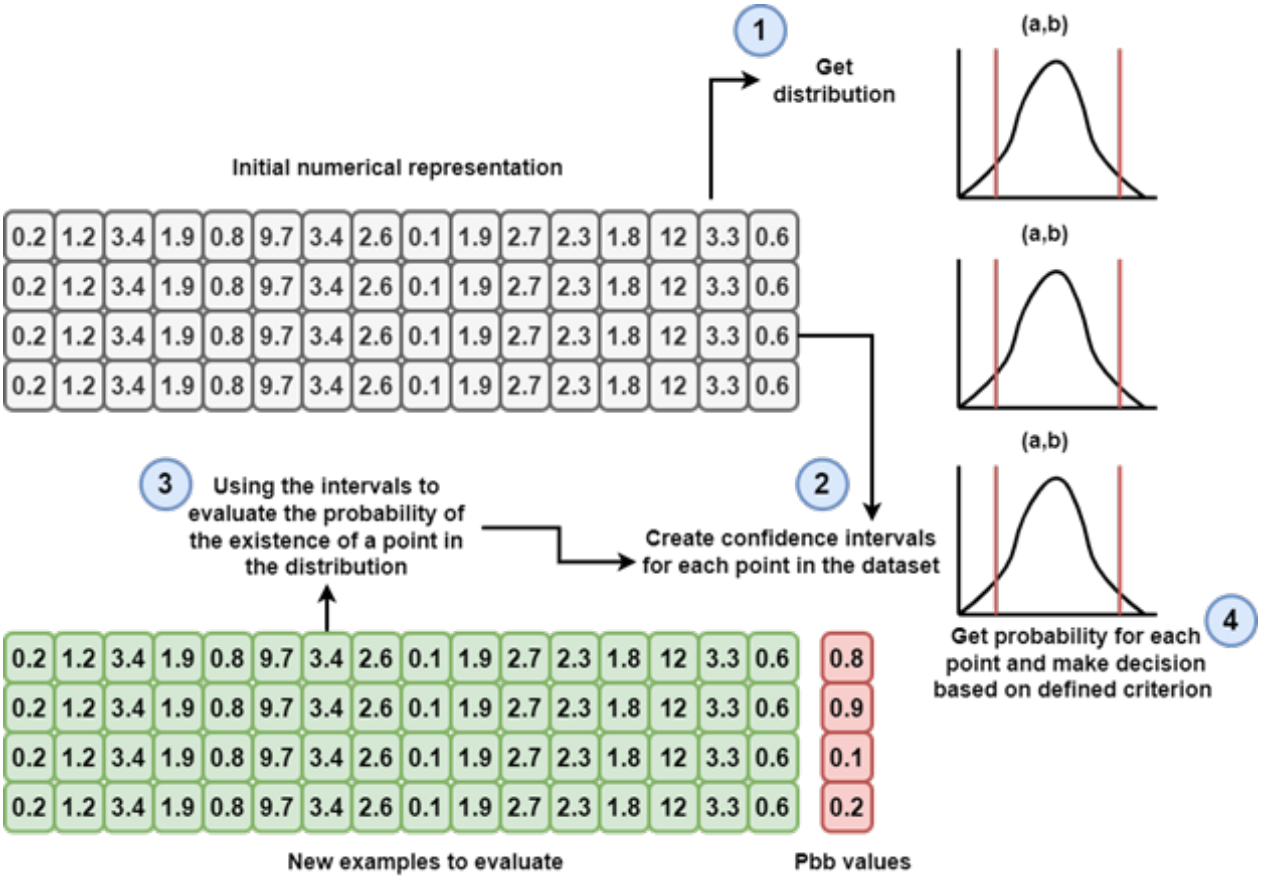


Figura 4.3: Aplicación de espacios latentes y estrategias de probabilidad para evaluar nuevas secuencias

4.2. Resultados y discusiones

4.2.1. Exploración de landscapes

Usando la metodología propuesta para la reconstrucción del *landscapes*, en una primera instancia se implementó un conjunto de datos de mutaciones puntuales de la proteína *Dihydrofolato reductase*, asociando la respuesta de interés a valores de $\Delta\Delta G$, los cuales permiten cuantificar la estabilidad térmica. Se recopilaron 556 mutaciones puntuales de estudios anteriores [18, 126, 9, 20]. De ellos, se seleccionaron aleatoriamente 400 ejemplos para generar un conjunto de datos de entrada, y aplicando la metodología propuesta en el capítulo anterior se

entrenaron modelos predictivos de regresión, logrando un desempeño de coeficiente de *Spearman's rank* de **0.76** (Ver Figura 4.4 A). Luego, se realizó una exploración de mutagénesis reconstruyendo el *landscape* para una mutación al mismo tiempo, con el fin de evaluar la estabilidad de la proteína, empleando el modelo predictivo ensamblado entrenado. De esta forma, se obtuvo el *landscape* mutacional de la proteína de interés.

Una vez obtenido el *landscape*, se clasificaron las mutaciones en tipos de estabilidad: creciente o decreciente, según el valor y el signo de $\Delta\Delta G$ [175]. Se evaluó la calidad de las predicciones utilizando las 156 entradas restantes no consideradas para el desarrollo del modelo ensamblado, obteniendo una precisión del **88.53 %** (Ver Figura 4.4 B).

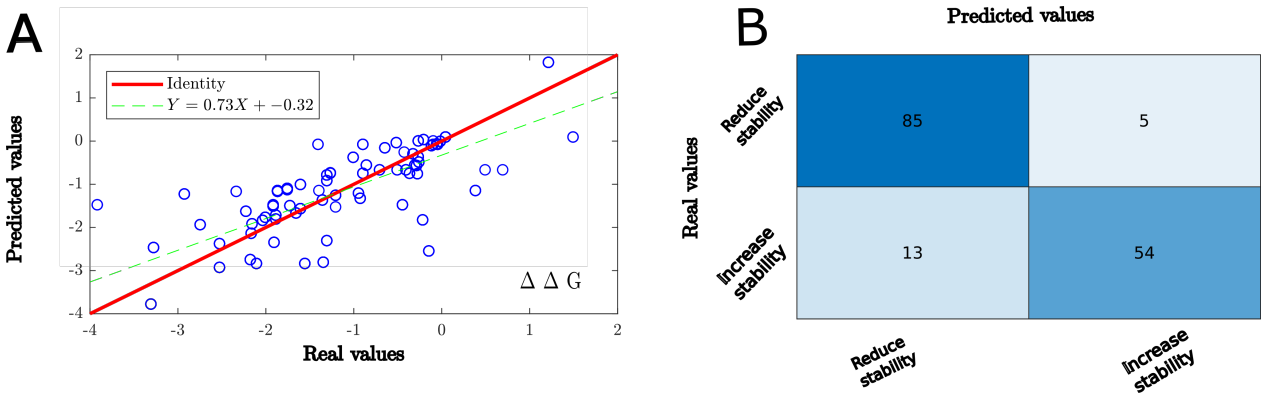


Figura 4.4: Rendimiento de modelos predictivos de estabilidad térmica para proteína *Dihydrofolate reductase*

Una vez obtenido el *landscape*, se analizó la distribución de $\Delta\Delta G$ en todas las posiciones para identificar mutaciones no informadas de interés potencial (Ver Figura 4.5 B y Ver Figura 4.5 A). Tal como se esperaba, un número reducido de mutaciones aumenta la estabilidad de la proteína sin ser perjudicial para la funcionalidad de la misma ($\Delta\Delta G \leq 3$, [175]).

Del *landscape* reconstruido, se seleccionó conjuntos de mutaciones no obvias, es decir, mutaciones no estrictamente relacionadas con el sitio de interacción de la proteína, de las cuales se obtuvieron las estructuras tridimensionales de cada variante usando el software SDM [121] y se aplicó un algoritmo de representación de proteínas como estructuras de grafos dirigidos basados en el desarrollo de estimadores de interacciones electrostáticas débiles, enfocados principalmente en los puentes de hidrógeno.

Al combinar los resultados obtenidos empleando la metodología propuesta con las representaciones basadas en estructuras de grafos, se identificaron patrones relacionados con enlaces de hidrógeno u otras interacciones electrostáticas débiles que explican los cambios de estabilidad (Ver Figura 4.6, en rojo se expone el cambio, mientras que las diferencias de las aristas indican si se ganan o pierden interacciones débiles). Se observa que los cambios de **ILE60PRO** y **PRO25GLU** fomentan la estabilidad de la proteína aumentando los puentes de hidrógenos. Por otro lado, los cambios de **VAL93PRO** y **SER3VAL** fomentan una disminución de la estabilidad, asociada a los cambios en las interacciones, ya sea aumentando perjudicialmente la estabilidad por medio de la generación de más puentes de hidrógeno o por medio de la desestabilización eliminando interacciones electrostáticas débiles.

De esta manera se demuestra que el método propuesto permite la simulación de variantes,

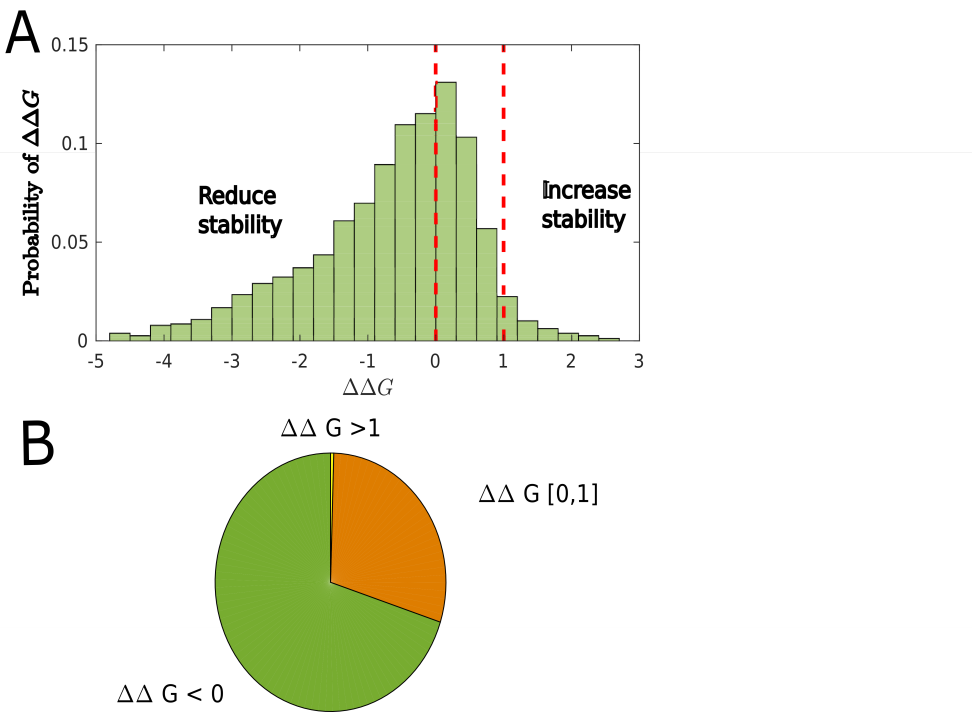


Figura 4.5: Resumen de tipos de mutaciones obtenidas a partir de la reconstrucción del *landscape* de *single point* para la proteína *Dihydrofolate reductase*

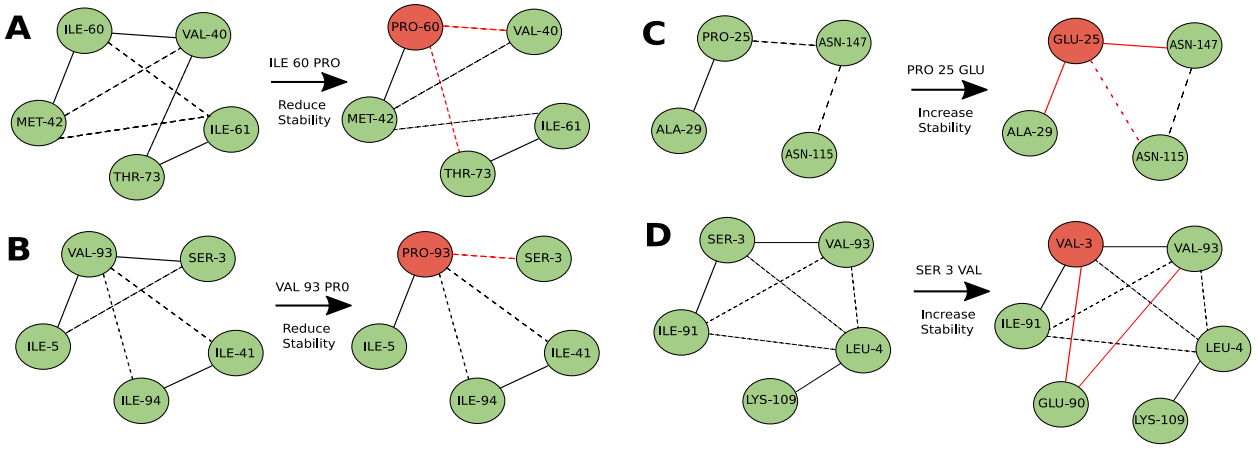


Figura 4.6: Representaciones de estructuras de grafos para la identificación de patrones relacionados con los cambios estructurales que influyen en la estabilidad de la proteína.

explorando un *landscape* de mutaciones, a partir del cual es factible identificar mutaciones que mejoren un fitness de interés, mostrando la utilidad del método propuesto para el diseño de variantes con propiedades deseables.

4.2.2. Exploración de actividades biológicas para secuencias de péptidos

Uno de los problemas más desafiantes en la ingeniería de proteínas se centra en diseñar proteínas con propiedades deseables [181]. Actualmente, existen dos estrategias de desarrollo, el diseño racional y la evolución dirigida [174]. El desarrollo de herramientas computacionales ha facilitado esta tarea, naciendo tópicos como el diseño semi-racional [157] y el *machine learning directed evolution* [174]. Sumados a esta iniciativa y contemplando las ventajas de la combinación de los codificadores semánticos desarrollados y la representación en el espacio de señales, se diseñó e implementó una metodología inicial basada en exploración por *fuerza bruta*, para el diseño de secuencias con propiedades deseables. En este caso, diseño de péptidos antimicrobianos.

Dado lo anterior, se entrenaron dos modelos predictivos usando la metodología propuesta en el capítulo anterior sobre el conjunto de secuencias procesadas. El primero es un modelo binario de clasificación de actividad antimicrobiana, mientras que el segundo corresponde a un modelo multi clase de variadas actividades biológicas para péptidos antimicrobianos, tales como antibacteriano, antiviral, anti-cáncer, anti-VIH y anti-fungal. El desempeño logrado fue de un **98.6 %** de precisión y **95.11 %**, respectivamente. Los altos valores de desempeño obtenidos se debe principalmente a la clara separación entre los péptidos antimicrobianos y no antimicrobianos (para el caso del clasificador binario) y patrones marcados para cada actividad biológica en el modelo multi clase (Ver Figura 4.7. Importante, se escalan los valores a una misma escala para poder representar los resultados, esto provoca que los patrones de espectro se aplanen, no visualizándose claramente los patrones en los péptidos con actividad antimicrobiana), lo cual, facilita el entrenamiento de modelos predictivos gracias a la identificación de patrones claros que permiten generalizar de mejor forma los comportamientos existentes en el conjunto de entrenamiento.

Una vez construidos los modelos, se exploraron nuevas secuencias siguiendo los siguientes pasos. i) Primero, se recolectaron diferentes secuencias peptídicas aleatoriamente desde la base de datos *PeptipediaDB* [130]. Notablemente, estas nuevas secuencias no se utilizaron durante la etapa de entrenamiento de modelos. En el caso de que no se tengan secuencias para explorar, se recomienda el uso de estrategias de *deep generative models* o similares [176] para crear nuevas secuencias. No obstante, esta estrategia no se aplicó en este trabajo debido a la cantidad de secuencias necesarias para generalizar los comportamientos de aprendizaje para el desarrollo de los *deep generative models*. ii) Se codificó y transformaron las secuencias utilizando la metodología propuesta en el capítulo 2 de esta tesis de doctorado. iii) Se evaluó la representación numérica estimando la probabilidad de que una nueva secuencia pertenezca al espacio latente de cada categoría usando la metodología propuesta en este capítulo. v) Se utilizó el modelo de entrenamiento para predecir la categoría de la secuencia propuesta. vi) Se evaluaron las predicciones, comprobando si las secuencias propuestas están clasificadas en la clase de interés.

Usando la estrategia planteada, se exploraron 10.000 secuencias aleatoriamente extraídas desde *PeptipediaDB* [130] y se definió un criterio de selección del 90 % de probabilidad de existir dentro del espacio latente conjunto para la actividad biológica deseable, en este caso, péptidos antimicrobianos y sus diferentes sub categorías. De las 10.000 secuencias exploradas,

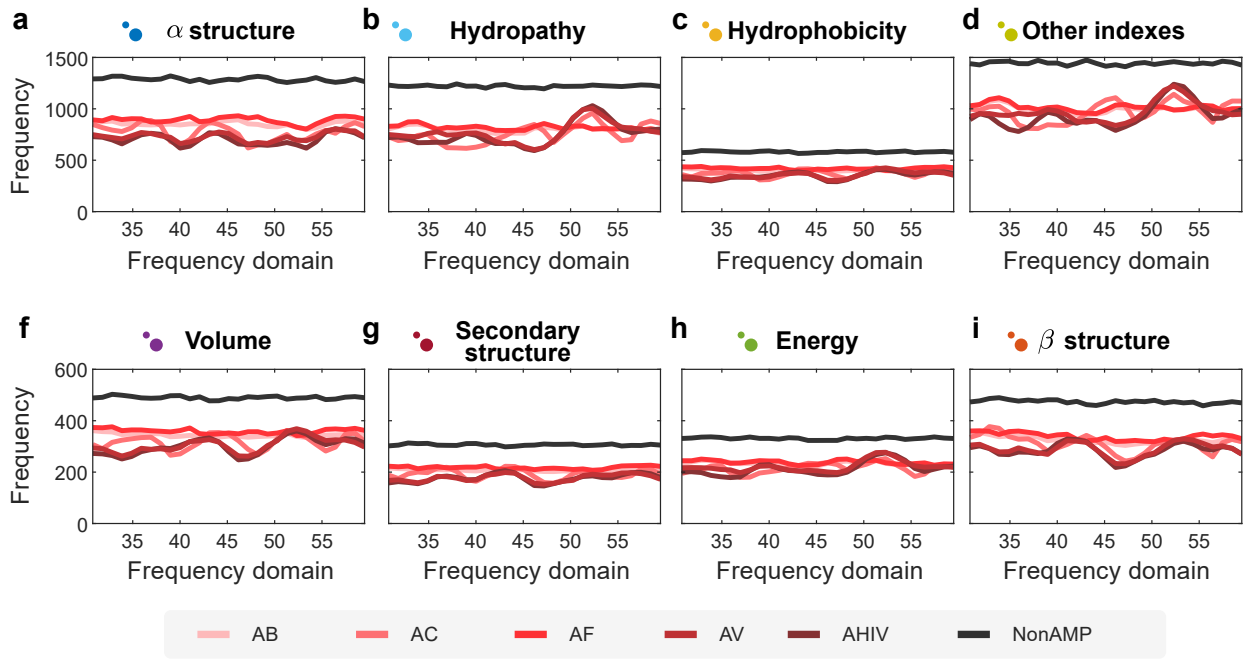


Figura 4.7: Patrones de señales para las secuencias de péptidos y las respectivas actividades biológicas analizadas

solo 3.513 cumplieron con el criterio de probabilidad establecido y se predijo su actividad usando los modelos previamente entrenados. Debido a que se conocían con antelación las secuencias, se pudo evaluar el rendimiento de la metodología de diseño comparando las clasificaciones predichas con las actividades biológicas reportadas por cada secuencia. La Tabla 4.1 muestra el desempeño obtenido en términos de rendimiento.

Actividad Biológica	**Número de secuencias**	**Accuracy (%)**
Antimicrobial	3513	98.1 %
Anti-viral	1210	91.4 %
Anti-bacterial	1350	89.3 %
Anti-cancer	1980	87.1 %
Anti-fungal	832	79.4 %
Anti-VIH	1127	82.3 %

Tabla 4.1: Rendimientos obtenidos para la estrategia de exploración de secuencias diseñada para este proceso

Notablemente, las secuencias reconocidas como péptidos antimicrobianos mostraron desempeño similar al resultado del entrenamiento. No obstante, el resto de las actividades biológicas evaluadas presentó una baja con respecto al modelo predictivo. Podemos inferir que esto es debido a que los modelos generados fueron entrenados con secuencias que solo presentaban una actividad en específica, mientras que las secuencias evaluadas presentaron en su mayoría actividad *moonlight* (razón por la cual las sub actividades de péptidos antimicrobianos no suman el total de secuencias evaluadas). Pese a estos resultados, la metodología propuesta facilita la exploración de nuevas secuencias desde un punto de vista probabilístico, siendo enormemente eficiente para péptidos antimicrobianos.

Tarea	Rendimiento	Entrenamiento	Validación	Tasa de sobreajuste
Productividad de péptidos	Accuracy	0.84	0.81	1.03
	Precision	0.89	0.89	1
	F1-Score	0.83	0.84	0.98
	Recall	0.81	0.88	0.92
Péptidos VIH, estimación de IC50	Accuracy	0.85	0.85	1
	Precision	0.81	0.83	0.98
	F1-Score	0.83	0.88	0.95
	Recall	0.83	0.88	0.95

Tabla 4.2: Resumen general, mejores desempeños obtenidos según diferentes métricas para cada tarea desarrollada

4.2.3. Diseño e implementación de modelos productivos para elaboración de péptidos con propiedades deseables

Finalmente, el último de los objetivos por abordar con respecto a los métodos de diseño se centra en la elaboración de sistemas predictivos para trabajar con sistemas poco informativos, es decir, con conjuntos de datos con un número de ejemplos menor a 50 registros (como en el caso de los sistemas predictivos de productividad para la maquinaria de *MachinePep*), mientras que para ambos, se remarca el hecho de la elaboración de métodos probabilísticos para estimar variables numéricas a partir de sistemas de clasificación categóricos.

Con base en lo anterior, se logró hacer representaciones numéricas de las secuencias de péptidos para ambos casos de estudio. Por otro lado, la categorización de los conjuntos de datos fue exitosa, logrando crear set de datos equilibrados con ausencia del desbalance de clases. Para el caso de la predicción de productividad en péptidos producidos en el sistema de *MachinePep*, dos fueron las categorías generadas, las cuales representaban *"Buena productividad"* y *"Mala productividad"*, mientras que para el caso de los péptidos con actividad anti-VIH fueron cuatro, definidas equitativamente mediante distribuciones cuartiles. Las cuales se definieron como

- C1: Muy Bueno
- C2: Bueno
- C3: Regular
- C4: Malos

Dado esto, las exploraciones de los modelos consideraron algoritmos clásicos del *machine learning*, variando sus hiperparámetros de configuración y midiendo su desempeño con las métricas clásicas. Recordando que los primeros modelos se basan en representaciones y *over sampling* empleando *data augmentation* con base en la combinación de las propiedades fisicoquímicas y contemplando que los seleccionados se emplearon para validación *leave one peptide out*, la Tabla 4.2 resume los desempeños logrados en la primera etapa del entrenamiento.

Tal como se observa en la Tabla 4.2, los resultados obtenidos bordean en promedio los 0.83 u 83 % de rendimiento en ambas tareas, lo cual es bastante significativo considerando

Tarea	Rendimiento	Entrenamiento	Validación	Tasa de sobreajuste
Productividad de péptidos	Accuracy	0.80	0.80	1
	Precision	0.81	0.82	0.99
	F1-Score	0.82	0.81	0.99
	Recall	0.80	0.81	0.99
Péptidos VIH, estimación de IC50	Accuracy	0.81	0.82	0.99
	Precision	0.81	0.81	1
	Recall	0.80	0.83	0.96
	F1-Score	0.82	0.82	1

Tabla 4.3: Resumen mejores desempeños según *leave one peptide out*

la cantidad de ejemplos existentes en el conjunto de datos de productividad. No obstante, cabe recalcar que esto representa a los conjuntos con *data augmentation*, razón por la cual, el desempeño real puede estar ofuscado.

Los modelos seleccionados en la etapa 1, en ambos casos, se asociaron a algoritmos del tipo ensamble, tales como *random forest* y *Adaboost*. Esto puede ser principalmente a la complejidad existente en la identificación de relaciones que permitan separar el conjunto en sus respectivas categorías definidas.

A partir de estos modelos, se aplicó la estrategia de validación con *leave one peptide out*, en la cual, los desempeños bajan. No obstante, se mantienen por sobre el 80 %, tal como se resumen en la Tabla 4.3.

Como era de esperarse, los resultados muestran una baja en el desempeño. Sin embargo, no es significativa. Además, se mejora la taza de sobre ajuste.

En ambos casos se seleccionan modelos basándose en el algoritmo *random forest* con un número de estimadores de 1500 y un criterio de función de ganancia de información *entropy*.

Ya con los modelos generados, se procedió a implementar los modelos probabilísticos y las estrategias de espacio latente, con el fin de diseñar métodos que permitan evaluar nuevas secuencias y emplear los modelos predictivos generados.

Finalmente, combinando este análisis de espacio latente, con el modelo predictivo, se puede obtener la categoría de la nueva secuencia y estimar la probabilidad de su valor numérico empleando distancias euclidianas con base en estos anchos de banda y las secuencias en general. Esquemáticamente, el uso de los modelos construidos y los diferentes componentes de la metodología propuesta se resumen en la Figura 4.8.

Siguiendo el pipeline propuesto, se exploraron diferentes secuencias mediante una exploración de *landscape*, de las cuales aquellas con mayor probabilidad de éxito se están evaluando experimentalmente para ambos casos. Con el fin de evaluar las secuencias desde un punto de vista estructural, solo para las más relevantes y a modo de *ejemplo juguete*, dado que es un protocolo que actualmente se encuentra en desarrollo por parte del equipo de trabajo, se empleó la herramienta *AlphaFold* para poder obtener modelos estructurales de los péptidos y estimar su factibilidad de desarrollo, junto con su estimación de error.

Solo a modo ilustrativo, en las Figuras 4.9 y 4.10 se muestran los resultados para una secuencia de prueba con una probabilidad del 0.98 % de tener un IC50 perteneciente a la

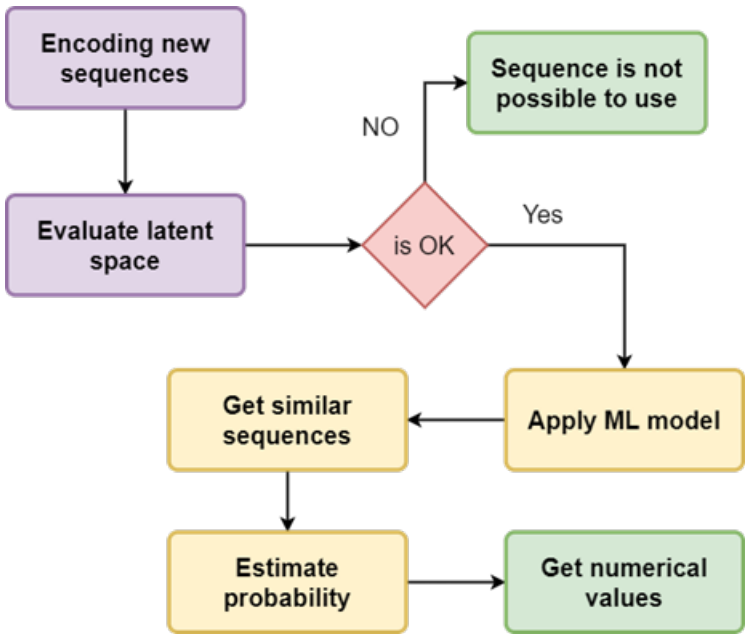


Figura 4.8: Pipeline propuesto para diseñar y explorar nuevas secuencias con nuevas secuencias.

categoría C1, exponiendo la estructura y el error de alineamiento previsto.

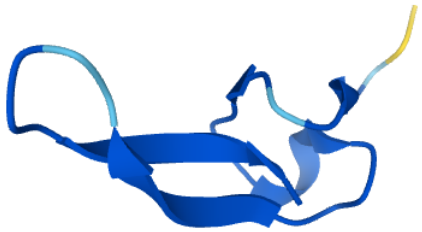


Figura 4.9: Modelo estructural obtenido con *AlphaFold* para secuencia diseñada por el método propuesto basado en exploración de espacio latente.

La Figura 4.9 muestra una estructura donde predominan más los dominios tipo β lo cual es característico en estructuras de péptidos, ya que se exhiben normalmente una mayor cantidad de residuos hidrofóbicos. Por otro lado, la mayoría de sus predicciones se encuentran dentro de una confidencia alta, lo cual denota que la calidad del modelo estructural generado es alta.

Los errores esperados visualizados en la Figura 4.10 son relativamente bajos, lo cual denota que las posiciones se encuentran dentro del margen aceptable según las indicaciones de los autores de *AlphaFold* [6].

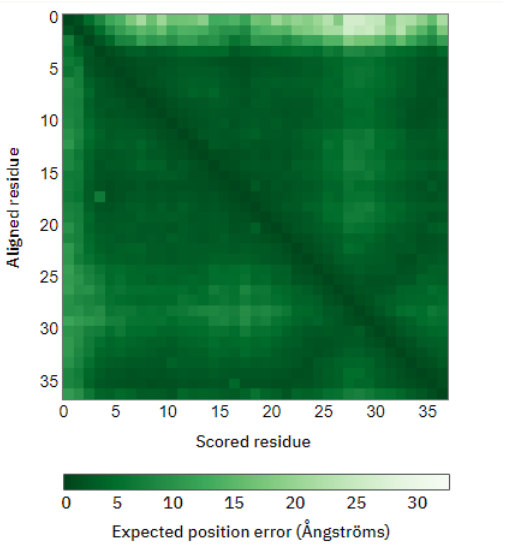


Figura 4.10: Error de alineamiento previo para el modelo generado, logrando que la mayoría de las posiciones se encuentren en un rango de error inferior a 5 Amstrong

4.3. Conclusiones y comentarios generales

El capítulo expuesto recientemente trató de exponer diferentes metodologías y estrategias computacionales para evaluar secuencias dentro de espacios latentes y facilitar herramientas que permitan diseñar o explorar secuencias con el fin de obtener propiedades deseables.

Bajo el margen de las metodologías diseñadas e implementadas, se pueden concluir los siguientes puntos más relevantes.

1. La estrategia de desarrollo de modelos predictivos ensamblados empleando los codificadores semánticos bajo representación de espacios de señales favoreció el diseño e implementación de modelos predictivos en diferentes ambientes de desarrollo y principalmente facilitó la exploración de *landscapes*.
2. La elaboración de técnicas de exploración y reconstrucción de fitness facilitó la identificación de mutaciones que incrementan la estabilidad para la proteína *dihydrofolate reductase*, logrando identificar mutaciones no obvias benéficas para la proteína. Esta estrategia junto con las representaciones de grafos favorece la identificación de patrones estructurales de una manera sencilla y eficiente.
3. El diseño e implementación de sistemas probabilísticos para la exploración de secuencias y la determinación de pertenencia a espacios latentes fue favorecido enormemente gracias a las representaciones basadas en transformadas de Fourier, lo cual da un punto de vista simple y comprensible a la hora de diseñar nuevas secuencias. El hecho de ocupar diferentes puntos de vista disminuye las probabilidades de error, ya que no es posible asegurar con los experimentos desarrollados que las funciones específicas tengan espectros específicos.
4. La implementación de estrategias de modelamiento predictivo para sistemas poco informativos soportados por técnicas de *data augmentation* demostró ser útil y lograr

desempeños eficientes, las cuales, combinadas con los métodos probabilísticos, facilitó dar una probabilidad de error a una predicción numérica relacionada con una previa clasificación, resultando ser una herramienta de utilidad al demostrar que las secuencias propuestas se modelan estructuralmente eficiente con *AlphaFold*.

Capítulo 5

Análisis de mutaciones puntuales e identificación de sitios relevantes para mutagénesis

El diseño de mutaciones puntuales es una de las temáticas de más interés en la ingeniería de proteínas, identificar sitios que afecten positivamente a una propiedad en particular en una proteína se ha convertido en el centro de los desarrollos de mutaciones sitio-dirigidas. Normalmente, las técnicas de diseño de mutaciones se han centrado en explorar o identificar variantes mutacionales que tiendan a mejorar la propiedad. En este sentido, la evolución dirigida se ha caracterizado por ir agregando cambios secuencialmente generación tras generación, mientras que el diseño racional se enfoca en el uso de los conocimientos sobre la proteína para poder proponer candidatos a mutar.

Ambas técnicas presentan problemáticas en términos de usabilidad y/o de conocimiento, razón por la cual, métodos computacionales se han diseñado e implementado con el fin de poder dar soporte a las metodologías clásicas de diseño, siendo el enfoque principal de los diseños semi-racionales. Bajo este alero, la identificación de los sitios importantes en una proteína juega un rol fundamental en el propósito de diseño.

Los principales métodos de estudio de sitios de interés en proteínas se han centrado en el componente termodinámico asociado a la estabilidad y los cambios que provoca mutar un residuo para la proteína en estudio, siendo una de las herramientas más empleadas para esto SDM [121], la cual se centra en medir los cambios de estabilidad y expresarlos en nivel de diferencia de energía libre entre los estados de la proteína ($\Delta\Delta G$). No obstante, no es la única herramienta de este tipo, el listado es tan largo como su historia, siendo las más relevantes, *Fold-X* [148], *Rosetta* [100] y *I-Mutant* [23].

Desde otro punto de vista, los esfuerzos se han centrado en la aplicación de técnicas basadas en el concepto filogenético y evolución de las proteínas. Una de las formas más interesantes, es estudiar las mutaciones y evaluar la probabilidad de sus cambios, contemplando que sitios conservados en las familias de proteínas tendrán baja probabilidad de mutar. Es más, en el caso de ocurrir, implicaría en efectos adversos a la función de la proteína, siendo este uno de los análisis que facilita la herramienta *MOSST* [118]. Sin embargo, la elabora-

ción de familias automáticas es compleja, debido a que involucran diversos conceptos que se asocian al conocimiento y la experiencia del investigador.

Siguiendo en la misma línea, uno de los conceptos que más fuerza ha adquirido el último tiempo, es la aplicación de modelos de epistasis al estudio de proteínas [73], el cual permite identificar cómo el cambio en una posición afecta a los residuos aledaños, no necesariamente sus vecindades, lo cual beneficia enormemente la identificación de sitios de interés y condiciona el desarrollo de mutagénesis no solo al sitio de interés, sino que se determina también su efecto sobre otros.

Por otro lado, otro de los problemas más interesantes se basa en el diseño e implementación de modelos predictivos para estudiar el efecto de las mutaciones. En capítulos previos, se analizó que los modelos predictivos basados en representaciones numéricas pueden presentar buen desempeño. No obstante, la probabilidad de sobre ajuste es alta, lo cual afecta negativamente a su usabilidad a largo plazo. Este defecto se debe principalmente a la poca variabilidad que existen en los vectores que representan las secuencias, ya que, al analizar las técnicas de codificación, para efectos prácticos es solo un cambio en una posición en particular. Esto último se traduce en, por ejemplo, para *one hot*, se *.^{en}ciende"* y se *.^{a}paga"* un punto en particular. Por otro lado, al emplear propiedades fisicoquímicas, el cambio se produce solo en una posición del vector numérico generado. Para efectos de representación real (vectores numéricos en el dominio real) las diferencias son un poco más visibles, tal es el caso de las aplicaciones de FFT o la representación por técnicas de aprendizaje de codificación basadas en NLP. No obstante, las diferencias siguen siendo sutiles, lo cual afecta negativamente tanto al modelo como a su posterior uso, en tareas fundamentales como la reconstrucción de *landscapes*.

Dado a lo anterior, y con el fin de poder presentar una alternativa de desarrollo de modelos predictivos y estudio de mutaciones puntuales, se propone en este capítulo el diseño e implementación de una metodología que facilite el entrenamiento de modelos predictivos en conjuntos de datos de mutaciones puntuales. Primero, se diseña una estrategia de caracterización de las mutaciones desde los puntos de vista termodinámicos, filogenéticos y estructurales, así como también considerando el *.^{a}mbiente"* de la mutación. Luego, siguiendo el mismo paradigma de diseño de modelos predictivos guiado por heurísticas de algoritmos genéticos, se desarrolla un *framework* de entrenamiento de modelos para estudiar efectos de mutaciones puntuales. Finalmente, se propone la identificación de sitios relevantes de la proteína combinando los puntos de vista filogenéticos y termodinámico, los cuales se proponen como sistema de estudio y aplicación para mutaciones sitio-dirigidas.

La metodología planteada se evaluó por etapas, primero, se determinó el poder de generalización de la estrategia de entrenamiento de modelos predictivos, empleando diferentes conjuntos de datos de experimentos de evolución dirigida, diseño de variantes, o métodos de *deep mutational scanning*, entrenando los modelos predictivos y comparando los desempeños con las estrategias clásicas, tanto en rendimiento en métricas como de evaluación del sobre ajuste. Por otro lado, la evaluación de la identificación de sitios relevantes se corrobora con dos conjuntos de mutaciones reportadas, el primero, en evaluación de fitness, donde los sitios identificados como importantes, están correlacionados positivamente con un aumento de la propiedad, mientras que en el segundo caso, se trabaja con conjuntos de datos relacionados con el análisis del efecto de enantioselectividad. Se descarta estudiar estabilidad y propie-

dades similares con la metodología propuesta, debido principalmente a que se ocupan como input para su desarrollo, lo cual sería contraproducente.

De esta forma, se diseña e implementa una metodología que facilita la identificación de sitios relevantes y predice eficientemente el efecto de mutaciones, la cual puede ser utilizada en variadas aplicaciones y, combinadas con técnicas de diseño actual de variantes, facilitaría guiar el desarrollo de mutaciones a los sitios importantes de estudio.

5.1. Metodología

A continuación, se describe los principales puntos de la metodología propuesta, contemplando tanto la etapa de caracterización, como los entrenamientos y los reconocimientos de sitios de interés en las proteínas estudiadas.

5.1.1. Caracterización de mutaciones en conjuntos de proteínas

La caracterización de las secuencias se basa en emplear diferentes herramientas computacionales previamente reportadas para obtener propiedades que representen a las mutaciones. La Figura 5.1 muestra un esquema representativo de la estrategia de caracterización.

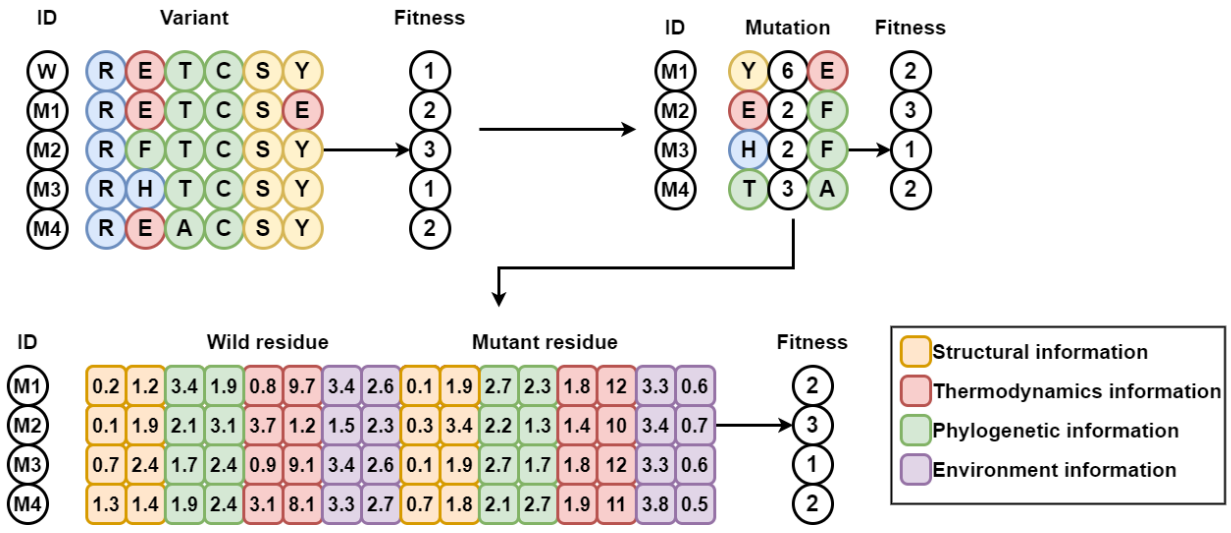


Figura 5.1: Esquema representativo de caracterización de conjuntos de datos de mutaciones puntuales

Tal como se nombró previamente, existen cuatro puntos de vista asociados a poder caracterizar las mutaciones, los cuales se describen a continuación.

El primero es el concepto termodinámico, para ello, se emplean la herramienta SDM [121] y se obtienen valores relacionados con los cambios de la estabilidad de la proteína, medidos en diferencia de energía libre $\Delta\Delta G$.

El segundo punto de interés es el asociado a las propiedades filogenéticas, para ello se emplean las herramientas *MOSST* [118] y *Evcoupling* [73] obteniendo de esta forma modelos estadísticos independientes para la clasificación de mutaciones como benéfica o perjudicial, así como también modelos epistáticos que facilitan la misma clasificación de mutaciones.

El tercer punto de interés se centra en la aplicación de los codificadores de propiedades fisicoquímicas y estructurales identificados en el capítulo 2 de este trabajo de tesis, con el fin de caracterizar a los residuos desde diferentes puntos de vista. Al usar este tipo de valores, se está adicionando información de carácter no lineal, debido principalmente a la forma en que se obtuvieron dichos codificadores.

Finalmente, el último punto de interés se centra en el estudio del ambiente, para ello, se emplea una ventana de tamaño 5, es decir, se contempla 5 residuos adicionales al residuo de interés, tanto a la derecha como a la izquierda, caracterizados por los tres puntos de vista expuestos previamente. La selección de este tamaño se basó principalmente en lo propuesto en [24].

5.1.2. Diseño y entrenamiento de modelos predictivos

El diseño e implementación de los modelos predictivos se basó en la metodología propuesta en el capítulo 3 de este trabajo de tesis, en el cual, se elaboró un *framework* de entrenamiento de modelos predictivos con base en técnicas de ensamble combinadas con métodos de optimización basados en algoritmos genéticos. Existen algunos puntos de interés, los cuales pueden ser resumidos a continuación.

- En este pipeline solo se exploran algoritmos de aprendizaje supervisado guiados por las heurísticas de algoritmos genéticos.
- Se hace una estandarización al conjunto de datos con el fin de disminuir el ruido que puedan presentar las variables.
- La división entre conjuntos de validación y entrenamiento es variable y depende del tamaño del conjunto de datos. No obstante, las proporciones son 80:20, 70:30 y 90:10.
- Al igual que en *framework* previamente desarrollado, se realiza la evaluación de sobre ajuste con técnicas de validación cruzada, mientras que la división inicial, se realiza 100 veces para dar soporte estadístico.

5.1.3. Identificación de sitios relevantes

La identificación de sitios relevantes en una proteína se propone contemplando los siguientes puntos de vista.

1. Mutaciones que según el modelo estadístico independiente obtenido por *MOSST* se clasifiquen como benéficas.

2. Mutaciones que según el modelo epistático obtenido por *Evcoupling* se clasifiquen como benéficas.

3. Mutaciones que según el punto de vista termodinámico se clasifiquen como estabilizadoras o sin efecto en la estabilidad.

Los puntos de vista anteriores pueden ser aplicados de manera inversa, esto aplicaría para identificar sitios que no deberían ser empleados para mutar, al menos por una de las tres condiciones.

Por otro lado, se aplica el modelo predictivo generado con el fin de explorar/reconstruir el *landscape* y predecir el efecto de la mutación, siendo el cuarto punto de interés aplicado para identificar sitios relevantes para mutaciones sitio-dirigidas.

5.1.4. Validación y conjuntos de datos de prueba

Con el fin de validar las estrategias planteadas, tanto en modelos predictivos como de identificación de sitios relevantes, se trabajan con diferentes conjuntos de datos, los cuales son obtenidos principalmente desde experimentos de evolución dirigida o exploraciones con *deep mutational scanning*, siendo previamente reportados en la literatura. La Tabla 5.1 resume los conjuntos empleados.

Tal como se puede observar en la Tabla 5.1, los conjuntos de datos estudiados corresponden a proteínas con variadas funcionalidades, propiedades y características. Todas las respuestas de dichos conjuntos se asocian al mismo tipo, en este caso fitness, razón por la cual no fue incluida en la tabla como columna adicional.

5.2. Resultados y discusiones

A continuación, se presentan los principales resultados obtenidos en la metodología propuesta, dividiéndolos entre desarrollo de modelos predictivos y evaluación de sitios relevantes identificados.

5.2.1. Entrenamiento de modelos predictivos para sistemas de mutaciones puntuales

Se entrenaron modelos predictivos siguiendo la metodología planteada en este capítulo para todos los conjuntos de datos resumidos en la Tabla 5.1. La Figura 5.2 muestra la medida de precisión obtenida no solo para los modelos entrenados empleando la metodología actual, sino que también compara los resultados con representaciones clásicas y con las representaciones y modelo predictivo desarrollado en el capítulo 3.

ID	Set de datos	Descripción	Referencias
UBE2I P63279	UBC9 HUMAN	Maquinaria central en la vía de sumoilación de la célula.	[171]
SUMO1 P63165	SUMO1 HUMAN	Participación en el proceso de ubiquitinación de proteínas dirigidas a la degradación proteasomal	[171]
TPK1 Q9H3S4	TPK1 HUMAN	Cataliza la conversión de tiamina en pirofosfato de tiamina, un cofactor de algunas enzimas de las vías glicolíticas y de producción de energía	[171]
CALM 1 P0DP23	CALM1 HUMAN	La calcodulina media el control de una gran cantidad de enzimas, canales iónicos, acuaporinas y otras proteínas mediante la unión al calcio	[171]
A0A221S5X8	Envelope protein E	La proteína E media la entrada del virus y la fusión de membranas y contiene los sitios de unión al receptor putativos para las células huésped	[152]
P60484	PTEN HUMAN	Fosfatasa dual con actividades tanto de proteína como de lípido fosfatasa	[137]
P51580	TPMT HUMAN	La tiopurina metiltransferasa metila compuestos de tiopurina	[137]
P46937	YAP1 HUMAN	Proteína que actúa como regulador transcripcional activando la transcripción de genes implicados en la proliferación celular y suprimiendo genes apoptóticos.	[137]
P28482	MK01 HUMAN	Actúan como punto de integración de múltiples señales bioquímicas, y están implicadas en una amplia variedad de procesos celulares, tales como proliferación celular, diferenciación celular, regulación de la transcripción y desarrollo	[137]
P61073	CXCR4 HUMAN	CXCR-4 es un receptor de alfa-quimiocinas específico para el factor 1 derivado del estroma (SDF-1 también llamado CXCL12)	[137]
P51681	CCR5 HUMAN	Receptor de quimiocinas en el grupo de quimiocina CC	[137]
P37231	PPARG HUMAN	Regula el almacenamiento de ácidos grasos y el metabolismo de la glucosa	[137]
Q6ZVK8	NUD18 HUMAN	Eliminan los metabolitos de nucleótidos potencialmente tóxicos de la célula y regulan las concentraciones y la disponibilidad de muchos sustratos de nucleótidos, cofactores y moléculas de señalización diferentes	[137]
BGLT3	Beta globin locus transcript 3	Regulador de la transformación celular mediada por BCR -ABL en la leucemia mieloide crónica	[137]

Tabla 5.1: Resumen de conjuntos de datos empleados para validar la metodología propuesta de identificación de sitios relevantes

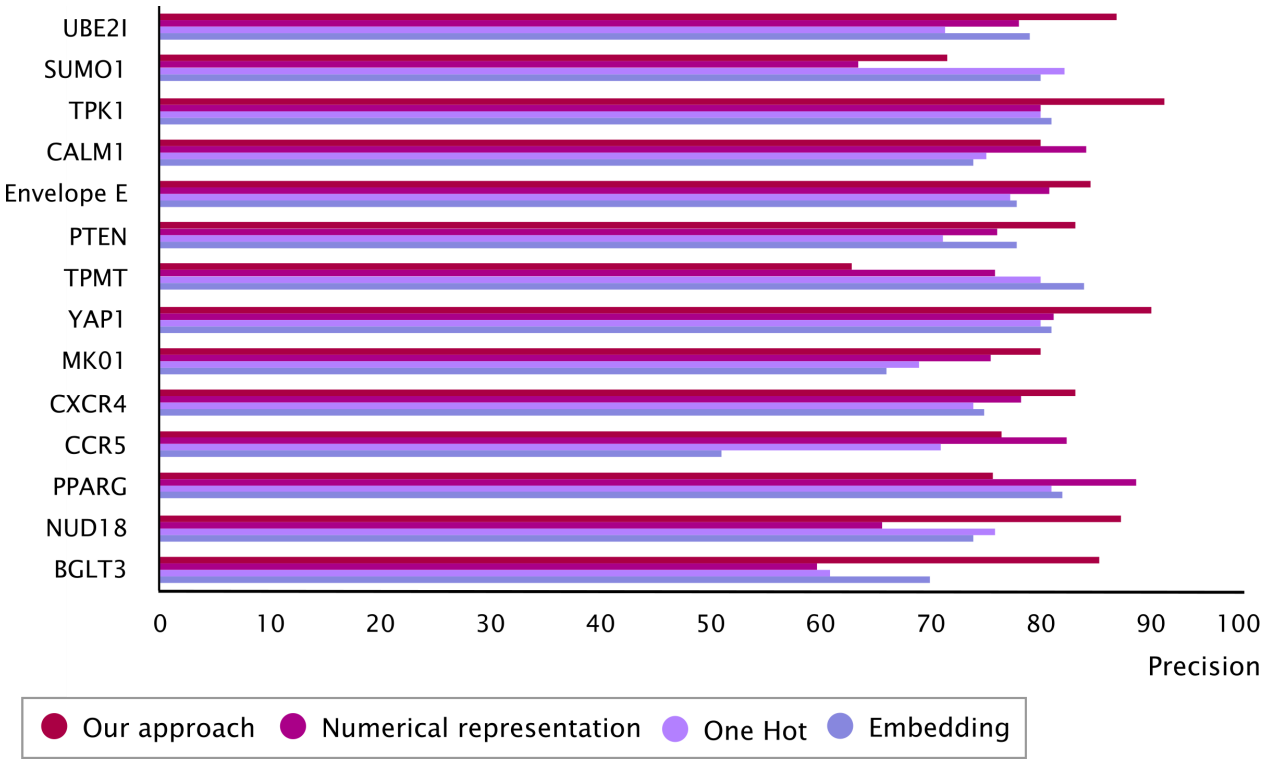


Figura 5.2: Comparación de medidas de desempeño para tareas evaluadas de sistemas de predicción de mutaciones puntuales

En 9 de 14 modelos la metodología de caracterización de secuencias logra mejores desempeños, teniendo en promedio una precisión del 81.4 %, mientras que en algunas ocasiones las representaciones por *embedding* o por *one hot* logran un mejor rendimiento, mientras que en el caso de las representaciones basadas en Fourier son semi estables. Sin embargo, a la hora de comparar el sobre ajuste, se observa que todas las estrategias presentan un mayor sobre ajuste en comparación a lo obtenido empleando el método planteado en este capítulo, lo cual es concordante con lo expuesto en el capítulo tres.

Esto indica que a pesar de que el desempeño disminuye en 5 de 14 modelos entrenados, la generalización que se logra es mayor, lo cual la hace más eficiente y por ende presenta un mejor desempeño, haciéndolo útil para variadas tareas de clasificaciones, predicciones o estudios de mutaciones puntuales.

5.2.2. Etapas y consideraciones al identificar sitios de interés

Los sitios de interés de una proteína se pueden ver desde diferentes puntos de vista, en este sentido, dependiendo del objetivo a trabajar, se proponen los siguientes puntos de interés.

- A nivel termodinámico, los sitios a identificar como posibles sitios de interés para mutagénesis sitio-dirigida, no deben alterar la estabilidad de la proteína, en otras palabras, la diferencia de energía libre entre el estado *wild type* y el mutado debe encontrarse en el rango de (-1 y 1) de $\Delta\Delta G$ [121].

- Tanto el modelo epistático como el modelo de *Evcoupling* lo deben clasificar como mutación benéfica.
- El modelo de epistásis no lo puede identificar como un sitio con altas dependencias o interacciones con otros sitios (alto valor de *coupling*), ya que indica que infiere o tiene relación con otros sitios al mismo tiempo, lo cual indica que un cambio en ellos afecta a todo el resto de la secuencia con quienes interactúan.
- Aplicar el modelo predictivo y reconstruir el *landscape* según corresponda, luego, correlacionar los sitios de interés con los sitios de relevancia detectados.

5.2.3. Identificación de sitios relevantes en Epoxide Hydrolase en estudios de enantioselectividad

Como ejemplo práctico se expone el análisis a las mutaciones de la enzima *Epoxide Hydrolase*, la cual se obtiene de información desde mutaciones reportadas obtenidas por evolución dirigida [182], a la cual se le está evaluando la enantioselectividad.

Siguiendo las reglas establecidas en la sección anterior, se realizó la identificación de los sitios relevantes, de los cuales se contemplan los componentes de evolución filogenética, además de su estabilidad, para clasificarlos como sitios benéficos. Estos sitios a la hora de hacer la reconstrucción de fitness fueron identificados y al ser mutados se predijeron como sitios que afectan positivamente a la respuesta de interés (enantioselectividad). La Figura 5.3 muestra la posición estructural de los principales sitios identificados para la enzima de interés.

En este caso, en particular, los sitios más relevantes corresponden a los residuos R380, D179, I193, S58 y F87 son propuestos para mutar y estos se correlacionan con los resultados de evolución dirigida para esta proteína [182] ya que estos sitios normalmente afectan positivamente a la respuesta de interés. La reconstrucción del *landscape* y la identificación de los sitios contemplando los diferentes puntos de vista, favorecieron la elección de ellos, y efectivamente tienen una relación positiva, lo cual reafirma la hipótesis planteada y demuestra el poder de la metodología propuesta.

La identificación de sitios no solo puede ocuparse para determinar los sitios relevantes para efectos de mutaciones benéficas, sino que también como identificación de sitios que no deben ser modificados debido a que afectan negativamente a la proteína. En este sentido, la Figura 5.4 muestra el grafo de *coupling* en la proteína *Epoxide Hydrolase* y un *zoom* a una cierta zona de interés.

Los nodos más marcados en azul denotan una fuerte tendencia a influir en otros, esto es, desde un punto de vista epistático, afectan a otros residuos en su comportamiento. A la hora de estudiar dichos residuos, la correlación que existe entre el efecto de enantioselectividad (la respuesta de interés en esta proteína de estudio) y estos sitios denotan que al mutarlos, por cualquier residuo, se afecta negativamente la respuesta, lo cual, tiene bastante sentido, ya que es un sitio no perteneciente al sitio activo. Pero, si juega un rol fundamental en la enzima.

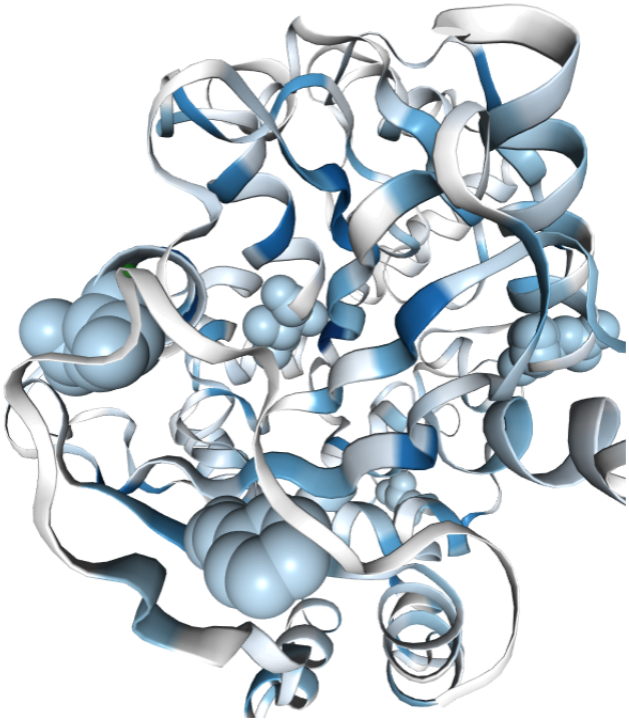


Figura 5.3: Sitios relevantes para mutagénesis dirigida según los criterios definidos la metodología propuesta

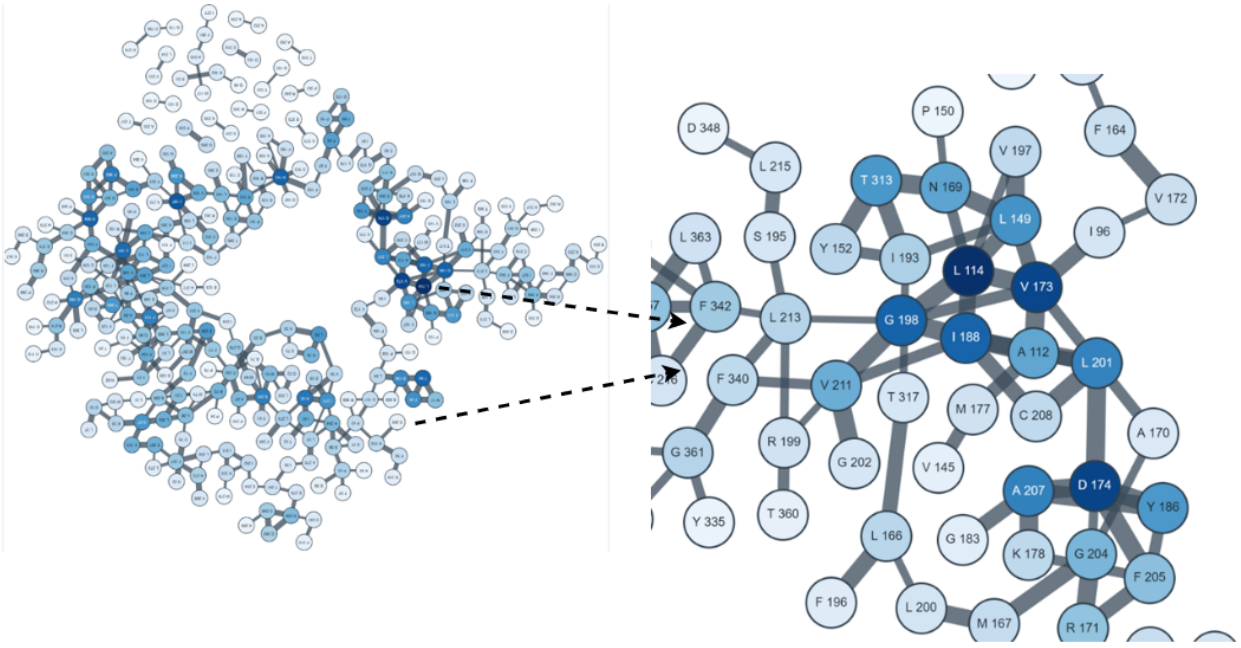


Figura 5.4: *Evolutionary coupling* aplicado a proteína *Epoxide Hydrolase* denotando la identificación de sitios que no deben ser mutados.

5.3. Conclusiones y comentarios generales

Como conclusiones, es posible comentar los siguientes puntos de interés.

1. La metodología de entrenamiento de modelos predictivos combinando las caracterizaciones desde diferentes puntos de vista facilitó el desarrollo de modelos predictivos eficientes, con un alto desempeño y un bajo sobre ajuste, en comparación a las estrategias clásicas y los métodos de representación de secuencias.

2. Se diseña e implementa una receta de puntos para identificar sitios de interés en una proteína, esta receta contempla los componentes filogenéticos, epistáticos y termodinámicos. Además, acoplados con la reconstrucción de *landscapes*, permite una identificación eficiente que puede correlacionar con las respuestas de interés.

3. Se demostró la usabilidad de la metodología propuesta estudiando la enzima *Epoxide Hydrolase* y los efectos de las mutaciones sobre la enantioselectividad de la enzima. Se muestra que no solo se identifican sitios de interés a mutagénesis que correlacionan con los valores mostrados por los autores [182]. Si no que también, permite la identificación de sitios relevantes para no mutar. Demostrando que la metodología planteada funciona como método de dirección de los procesos de diseño de mutaciones o evolución de las proteínas.

Capítulo 6

Conclusiones, trabajos en desarrollo y perspectivas a futuro

Una vez desarrolladas las diferentes estrategias computacionales propuestas y validadas en diferentes casos de estudio, es posible hacer un análisis global de ellas, desde una retrospectiva de funcionamiento, aplicaciones y el aporte que esto significa para la comunidad científica. Contemplando esto, a continuación, se presentan las conclusiones generales y las perspectivas a futuro, donde se exponen diferentes preguntas que pueden ser abordadas como continuación de este trabajo de tesis, así como también el trabajo en desarrollo que se está llevando a cabo.

6.1. Conclusiones generales

Como conclusiones generales se pueden mencionar los siguientes puntos.

1. Se identificaron grupos semánticos de propiedades fisicoquímicas combinando técnicas de *text mining* con algoritmos de aprendizaje no supervisado, estás propiedades son representadas como componentes no lineales mediante kernel-PCA y pueden ser empleadas como propiedades para caracterizar mutaciones. Remarcablemente, su uso como método de representación numérica de secuencias, combinado con transformadas al espacio de señales, facilita el entrenamiento de modelos predictivos, brindando información suficiente para mejorar el rendimiento predictivo y los índices de sobre ajuste. Además, dichas representaciones favorecen la identificación visual de patrones y fomentan separaciones marcadas relacionadas con la función de la proteína y su plegamiento.
2. Se diseñó e implementó un *framework* de entrenamiento de modelos predictivos, el cual combina estrategias de optimización heurísticas basadas en algoritmos genéticos con algoritmos de aprendizaje supervisado, enmarcado en un sistema de aprendizaje ensamblado, empleando como input las representaciones de espacios de frecuencia basadas en los grupos de propiedades fisicoquímicas identificadas. Por otro lado, los resultados muestran no solo un aumento en el desempeño de los modelos predictivos, sino que también una disminución en el sobre ajuste para variadas tareas evaluadas, lo

cual denota no solo una generalización en el comportamiento del aprendizaje, sino una generalización desde el punto de vista de un método de entrenamiento.

3. Se diseñaron e implementaron estrategias de exploración de variantes, permitiendo la reconstrucción de *landscapes* para fitness específicos, evaluación estadística de *landscapes* de representaciones de Fourier basados en propiedades fisicoquímicas y la elaboración de sistemas predictivos estadísticos para dar soporte a la evaluación de nuevas variantes en un ámbito de diseño de secuencias. Además, las metodologías propuestas constituyen un avance en términos de facilitar un método de trabajo para conjuntos de datos poco informativos, lo cual se solventa mediante la aplicación bien definida de *data augmentation*. Notablemente, la combinación de las representaciones propuestas junto con el *framework* de entrenamiento facilitan una exploración variada y eficiente de variantes mutacionales para tareas particulares o propiedades deseables.

4. La combinación de los puntos de vista termodinámicos, filogenético y estructural facilitó el diseño e implementación de estrategias para el entrenamiento de modelos proteína-respuesta específicos en conjuntos de datos de mutaciones puntuales. Estos modelos facilitan la exploración de *landscapes*, presentan un desempeño eficiente y un sobre ajuste mínimo para la mayoría de los casos evaluados, en comparación a las representaciones clásicas empleadas a modo de comparación de resultados. Además, estos mismos criterios facilitan la identificación de sitios relevantes para mutaciones sitio-dirigidas en proteínas, lo cual permite guiar la evolución de las proteínas para obtener mutaciones con propiedades deseables y optimizar el funcionamiento correspondiente, convirtiéndola en una herramienta relevante que al incorporarlas a los métodos actuales de diseño fomentaría un mejor rendimiento y una disminución en costo experimental.

6.2. Trabajo en desarrollo y resultados preliminares

Actualmente, se está trabajando en la elaboración de una suite computacional que facilite la aplicación de las metodologías planteadas para científicos sin las competencias informáticas necesarias para su implementación. Además, se está elaborando un protocolo que permita incorporar y simular la evolución dirigida guiada por los sitios de interés reconocidos en cada iteración, con el fin de identificar trayectorias, las cuales pueden ser aplicadas como inputs para entrenamiento de modelos predictivos y reconstrucción de etapas de diseño de mutaciones en proteínas de interés.

Como trabajo en desarrollo, también se está diseñando e implementando una estrategia computacional para la identificación de patrones en proteínas, la cual se centra en la utilización de las representaciones de secuencias combinadas con propiedades filogenéticas para estimar relaciones de parentesco entre las secuencias. La Figura 6.1 muestra un esquema representativo de la metodología propuesta.

Esta metodología contempla la incorporación de información desde diferentes puntos de vista, lo cual fomenta un mayor poder de decisión para separar los grupos. Los pasos de la estrategia propuesta se resumen a continuación.

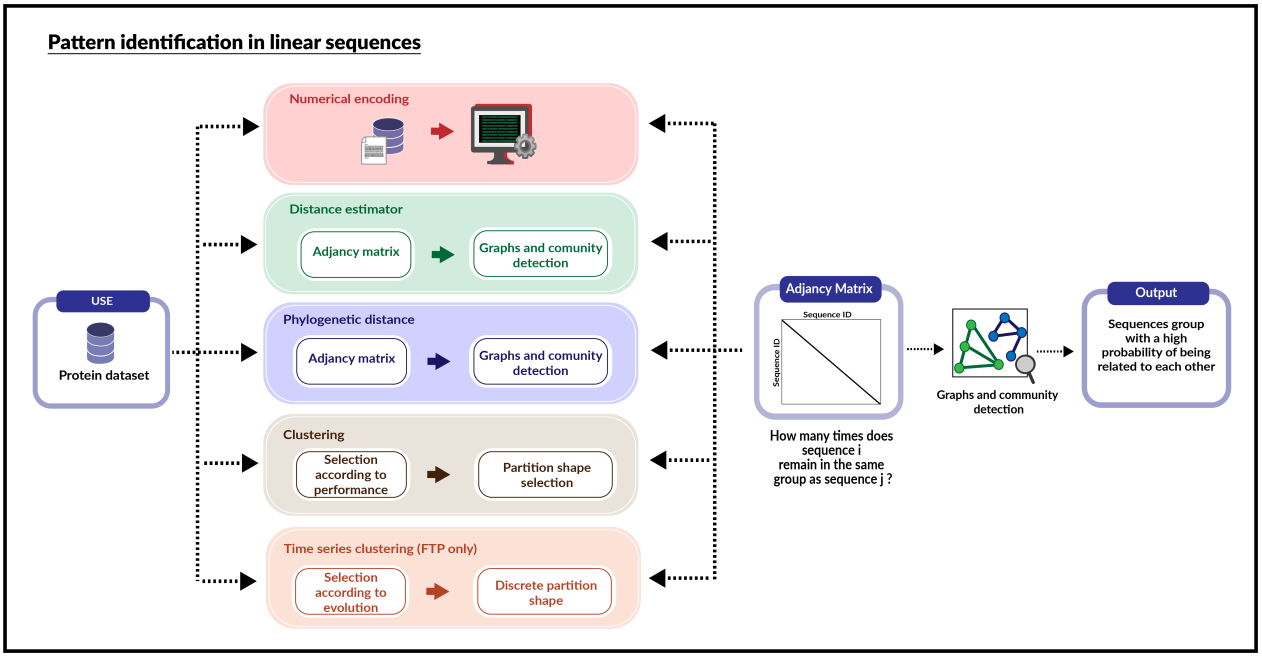


Figura 6.1: Esquema representativo de metodología de identificación de patrones actualmente en desarrollo.

1. Representar numéricamente las secuencias contemplando las propiedades identificadas en el capítulo dos.
2. Estimar la distancia de las secuencias a nivel numérico, contemplando las variaciones tanto en la forma de la curva como en el área bajo ella.
3. Estimar distancias filogenéticas mediante alineamientos múltiples de secuencias.
4. Armar matriz de adyacencia y estructura de grafos para aplicar clustering por medio de la detección de comunidades, para cada una de las representaciones generadas.
5. Solo en el caso de las representaciones por transformadas de Fourier, aplicar exploración de clustering de series de tiempo.
6. Generar una matriz de relaciones de secuencias que facilite la cuantificación de las relaciones entre secuencias, esto es, para una secuencia S_i cuántas veces se identificó en el mismo grupo que la secuencia S_j.
7. Una vez construida la matriz, desarrollar una matriz de probabilidad de relaciones y representarlas en estructuras de grafos.
8. Aplicar algoritmo de comunidades para la identificación de grupos de secuencias relacionadas.

La metodología expuesta ha sido testeada en diferentes conjuntos de datos relacionados con tareas de ingeniería de proteínas, contemplando proteínas de interacción a DNA, actividades biológicas de péptidos y clasificación de familias de hidrofobinas. Se compararon los resultados con herramientas clásicas de clustering de secuencias. Los resultados obtenidos a la fecha se muestran en la Figura 6.2.

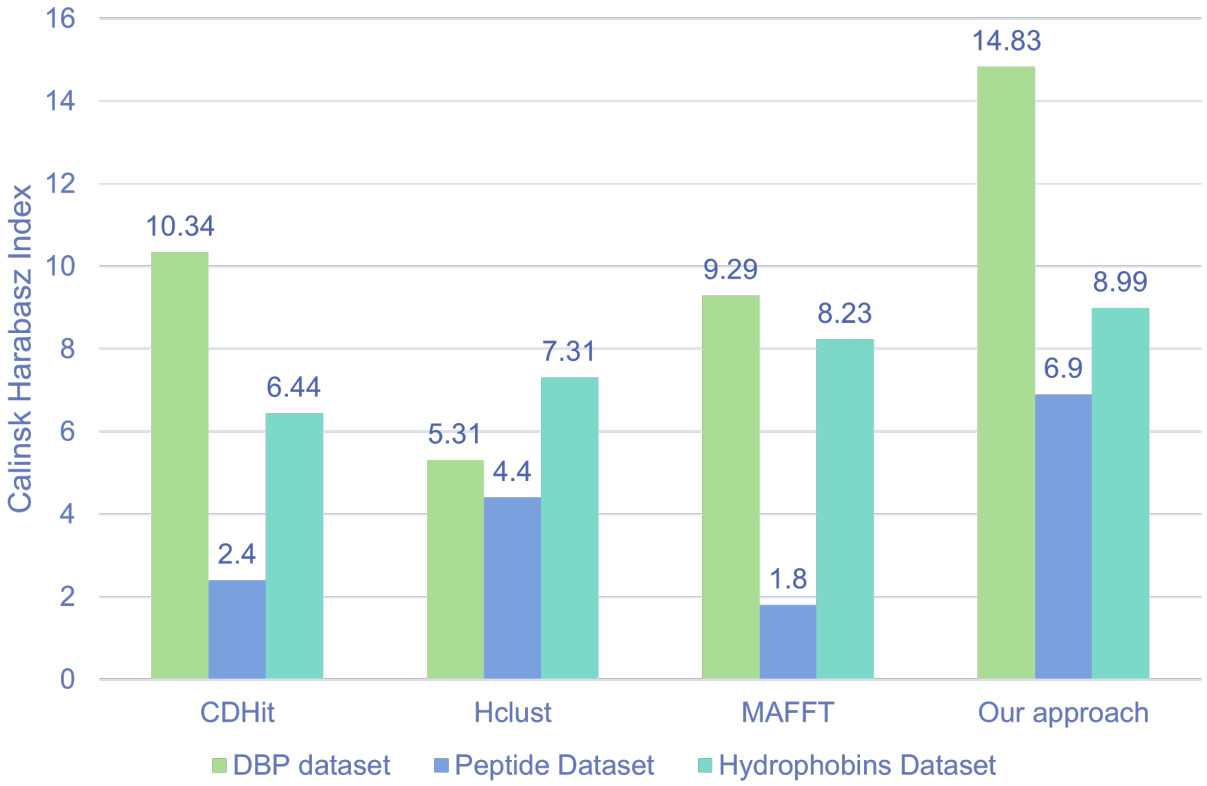


Figura 6.2: Resultados preliminares de algoritmo de clustering de secuencias de proteínas.

Los resultados obtenidos se comparan con las herramientas clásicas como *CDHit* [59], *Hclust* [113] y *MAFFT* [81], empleando para ello la métrica de *Calinski-Harabasz index*, mostrando un mejor rendimiento para los conjuntos de datos evaluados para la metodología de clustering que se está proponiendo.

Actualmente, esta metodología está siendo implementada bajo un sistema de paralelización con el fin de disminuir costo computacional. Además, de ser parametrizable eficientemente e incorporar algoritmos de heurística para optimizar la hiperparametrización de los algoritmos de *machine learning* incluidos en el proceso.

6.3. Perspectivas y proyectos a futuro

Con respecto a las perspectivas y proyectos a futuro, ya concluida esta tesis de doctorado, se pueden mencionar los siguientes aspectos de interés.

1. Emplear las metodologías propuestas para estudiar complejos de interacción proteína-proteína y proteína-ligando y la elaboración de estrategias computacionales para representar numéricamente los ligandos del tipo no proteico.
2. Diseñar e implementar estrategias que fomenten el uso de convoluciones para trabajar con CNN y métodos de *deep learning*, teniendo como base las representaciones de

secuencias basadas en espectros de frecuencias, lo cual debería facilitar la identificación de patrones no obvios en familias de proteínas.

3. Un análisis completo de todas las secuencias de *Uniprot/PDB* para caracterizarlas e identificar patrones desde un punto de vista de espectros de frecuencias, con el fin de determinar fehacientemente, los espectros que las identifican, y de esa forma, elaborar sistemas de anotación de proteínas con base en sus espectros de frecuencia, desarrollando un estudio completo de proteomas.

4. Aplicaciones a problemas específicos de ingeniería de proteínas, por ejemplo, el estudio del efecto clínico de mutaciones asociadas al cáncer, enfermedades autoinmunes o problemas genéticos, con el fin de identificar los patrones de interés y fomentar el desarrollo de un test de diagnóstico preventivo.

5. Incorporar las metodologías propuestas a los métodos de diseño en aplicaciones experimentales y corroborar los resultados, con el fin de aplicar técnicas de aprendizaje reforzado, que permitan penalizar las acciones de error.

Bibliografía

[1] Ahmed Abdelaziz, Mohamed Elhoseny, Ahmed S. Salama, and A.M. Riad. A machine learning model for improving healthcare services on cloud computing environment. *Measurement*, 119:117 – 128, 2018.

[2] Sheikh Adilina, Dewan Md Farid, and Swakkhar Shatabda. Effective dna binding protein prediction by using key features via chou's general pseaac. *Journal of Theoretical Biology*, 460:64 – 78, 2019.

[3] Talha Burak Alakus and Ibrahim Turkoglu. Prediction of protein-protein interactions with lstm deep learning model. In *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–5. IEEE, 2019.

[4] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

[5] Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 579–586. Association for Computational Linguistics, 2005.

[6] Mohammed AlQuraishi. Alphafold at casp13. *Bioinformatics*, 35(22):4862–4865, 2019.

[7] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[8] François Ancien, Fabrizio Pucci, Maxime Godfroid, and Marianne Rooman. Prediction and interpretation of deleterious coding variants in terms of protein structural stability. *Scientific reports*, 8(1):4480, 2018.

[9] François Ancien, Fabrizio Pucci, Maxime Godfroid, and Marianne Rooman. Prediction and interpretation of deleterious coding variants in terms of protein structural stability. *Scientific Reports*, 8(1):4480, 2018.

[10] Frances H Arnold. Design by directed evolution. *Accounts of chemical research*, 31(3):125–131, 1998.

[11] Minkyung Baek and David Baker. Deep learning and protein structure modeling. *Nature methods*, 19(1):13–14, 2022.

[12] Krishnan Balasubramanian and Satya P Gupta. Quantum molecular dynamics, topological, group theoretical and graph theoretical studies of protein-protein interactions. *Current topics in medicinal chemistry*, 19(6):426–443, 2019.

[13] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7):e0180944, 2017.

[14] Michael JA Berry and Gordon S Linoff. *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons, 2004.

[15] Maxwell L Bileschi, David Belanger, Drew H Bryant, Theo Sanderson, Brandon Carter, D Sculley, Alex Bateman, Mark A DePristo, and Lucy J Colwell. Using deep learning to annotate the protein universe. *Nature Biotechnology*, pages 1–6, 2022.

[16] Surojit Biswas, Grigory Khimulya, Ethan C Alley, Kevin M Esvelt, and George M Church. Low-n protein engineering with data-efficient deep learning. *Nature Methods*, 18(4):389–396, 2021.

[17] Brigitte Boeckmann, Amos Bairoch, Rolf Apweiler, Marie-Claude Blatter, Anne Estreicher, Elisabeth Gasteiger, Maria J Martin, Karine Michoud, Claire O'donovan, Isabelle Phan, et al. The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic acids research*, 31(1):365–370, 2003.

[18] A. J. Bordner and R. A. Abagyan. Large-scale prediction of protein geometry and stability changes for arbitrary single point mutations. *Proteins: Structure, Function, and Bioinformatics*, 57(2):400–413, Nov 2004.

[19] D. Braha and A. Shmilovici. Data mining for improving a cleaning process in the semiconductor industry. *IEEE Transactions on Semiconductor Manufacturing*, 15(1):91–101, Feb 2002.

[20] Aron Broom, Zachary Jacobi, Kyle Trainor, and Elizabeth M Meiering. Computational tools help improve protein stability but with a solubility tradeoff. *Journal of Biological Chemistry*, 292(35):14349–14361, 2017.

[21] Jason Brownlee. Why one-hot encode data in machine learning, 2017.

[22] Frédéric Cadet, Nicolas Fontaine, Iyanar Vetrivel, Matthieu Ng Fuk Chong, Olivier Savriama, Xavier Cadet, and Philippe Charton. Application of fourier transform and proteochemometrics principles to protein engineering. *BMC bioinformatics*, 19(1):382, 2018.

[23] Emidio Capriotti, Piero Fariselli, and Rita Casadio. I-mutant2. 0: predicting stability changes upon mutation from the protein sequence or structure. *Nucleic acids research*, 33(suppl_2):W306–W310, 2005.

[24] Emidio Capriotti, Piero Fariselli, and Rita Casadio. I-mutant2.0: predicting stability changes upon mutation from the protein sequence or structure. *Nucleic Acids Res*, 33(Web Server issue):W306–W310, Jul 2005.

[25] Emidio Capriotti, Piero Fariselli, Ivan Rossi, and Rita Casadio. A three-state prediction of single point mutations on protein stability changes. *BMC bioinformatics*, 9(2):S6, 2008.

[26] John F Carpenter, Michael J Pikal, Byeong S Chang, and Theodore W Randolph. Rational design of stable lyophilized protein formulations: some practical advice. *Pharmaceutical research*, 14(8):969–975, 1997.

[27] Petr Carsky and Miroslav Urban. *Ab initio calculations: methods and applications in chemistry*, volume 16. Springer Science & Business Media, 2012.

[28] David A Case, Thomas E Cheatham III, Tom Darden, Holger Gohlke, Ray Luo, Kenneth M Merz Jr, Alexey Onufriev, Carlos Simmerling, Bing Wang, and Robert J Woods. The amber biomolecular simulation programs. *Journal of computational chemistry*, 26(16):1668–1688, 2005.

[29] Jagat S Chauhan, Nitish K Mishra, and Gajendra PS Raghava. Prediction of gtp interacting residues, dipeptides and tripeptides in a protein from its evolutionary information. *BMC bioinformatics*, 11(1):1–9, 2010.

[30] Chi Chen, Ziheng Lu, and Francesco Ciucci. Data mining of molecular dynamics data reveals li diffusion characteristics in garnet li7la3zr2o12. *Scientific Reports*, 7:40769 EP –, Jan 2017. Article.

[31] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang. Disease prediction by machine learning over big data from healthcare communities. *IEEE Access*, 5:8869–8879, 2017.

[32] Wei Chen, Hui Ding, Pengmian Feng, Hao Lin, and Kuo-Chen Chou. iacp: a sequence-based tool for identifying anticancer peptides. *Oncotarget*, 7(13):16895, 2016.

[33] Chen-Fu Chien and Li-Fei Chen. Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry. *Expert Systems with Applications*, 34(1):280–290, 2008.

[34] Gobinda G Chowdhury. Natural language processing. *Annual review of information science and technology*, 37(1):51–89, 2003.

[35] Alexander P Christensen, Luis Eduardo Garrido, and Hudson Golino. Comparing community detection algorithms in psychological data: A monte carlo simulation. 2021.

[36] James W Cooley, PAW Lewis, and PD Welch. The fast fourier transform algorithm: Programming considerations in the calculation of sine, cosine and laplace transforms. *Journal of Sound and Vibration*, 12(3):315–337, 1970.

[37] Robert Cooley, Bamshad Mobasher, Jaideep Srivastava, et al. Web mining: Information and pattern discovery on the world wide web. In *ictai*, volume 97, pages 558–567, 1997.

[38] Irena Cosic. Macromolecular bioactivity: is it resonant interaction between macromolecules?-theory and applications. *IEEE Transactions on Biomedical Engineering*, 41(12):1101–1114, 1994.

[39] Irena Cosic, Drasko Cosic, and Katarina Lazar. Analysis of tumor necrosis factor function using the resonant recognition model. *Cell biochemistry and biophysics*, 74(2):175–180, 2016.

[40] Irena Ćosić and Dobrila NESIC. Prediction of 'hot spots' in sv40 enhancer and relation with experimental data. *European journal of biochemistry*, 170(1-2):247–252, 1987.

[41] Zhong Cui-xiang, Han Guo-qiang, and Huang Ming-He. Some new parallel fast fourier transform algorithms. In *Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05)*, pages 624–628. IEEE, 2005.

[42] Agata Czerniecka, Dorota Bielinska-Waz, Piotr Waz, and Tim Clark. 20d-dynamic representation of protein sequences. *Genomics*, 107(1):16–23, 2016.

[43] Chetna Dabas, Gaurav Kumar Nigam, and Himanshu Nagar. Investigating large-scale graphs for community detection. In *Innovations in Computational Intelligence and Computer Vision*, pages 122–129. Springer, 2021.

[44] Bihter Das and Suat Toraman. Classifying protein sequences using convolutional neural network. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 9:1663–1671, 12 2020.

[45] Joshua V Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

[46] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14(2):241–258, 2020.

[47] Pavlo O Dral. Quantum chemistry in the age of machine learning. *The journal of physical chemistry letters*, 11(6):2336–2347, 2020.

[48] L. Duan, W. N. Street, and E. Xu. Healthcare information systems: data mining methods in the creation of a clinical recommender system. *Enterprise Information Systems*, 5(2):169–181, 2011.

[49] Margaret H Dunham. *Data mining: Introductory and advanced topics*. Pearson Education India, 2006.

[50] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Computer Vision — ECCV 2002*, pages 97–112, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[51] Girum Fitihamlak Ejigu and Jaehee Jung. Review on the computational genome annotation of sequences obtained by next-generation sequencing. *Biology*, 9(9):295, 2020.

[52] Asif Ekbal, Sriparna Saha, Pushpak Bhattacharyya, et al. A deep learning architecture for protein-protein interaction article identification. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 3128–3133. IEEE, 2016.

[53] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: towards cracking the language of life's code through self-supervised learning. *bioRxiv*, pages 2020–07, 2021.

[54] Narayanan Eswar, Ben Webb, Marc A Marti-Renom, MS Madhusudhan, David Eramian, Min-yi Shen, Ursula Pieper, and Andrej Sali. Comparative protein structure modeling using modeller. *Current protocols in bioinformatics*, 15(1):5–6, 2006.

[55] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37, 1996.

[56] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34, 1996.

[57] Usama M Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, et al. Knowledge discovery and data mining: Towards a unifying framework. In *KDD*, volume 96, pages 82–88, 1996.

[58] Majid Forghani and Rouhollah Khani. A multivariate clustering of aaindex database for protein numerical representation. In *2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)*, pages 1–4. IEEE, 2017.

[59] Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu, and Weizhong Li. Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, 2012.

[60] Catherine Gallou, Claudine Junien, Dominique Joly, Frédéric Staroz, Marie Thérèse Orfanelli, and Christophe Béroud. Software and database for the analysis of mutations in the VHL gene. *Nucleic Acids Research*, 26(1):256–258, 01 1998.

[61] Thomas Gaudelet, Ben Day, Arian R Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu, Jeremy BR Hayter, Richard Vickers, Charles Roberts, Jian Tang, et al. Utilizing graph machine learning within drug discovery and development. *Briefings in bioinformatics*, 22(6):bbab159, 2021.

[62] Z. Ge, Z. Song, S. X. Ding, and B. Huang. Data mining and analytics in the process industry: The role of machine learning. *IEEE Access*, 5:20590–20616, 2017.

[63] Jordan Graves, Jacob Byerly, Eduardo Priego, Naren Makkapati, S Vince Parish, Brenda Medellin, and Monica Berrondo. A review of deep learning methods for antibodies. *Antibodies*, 9(2):12, 2020.

[64] Jonathan Greenhalgh, Apoorv Saraogee, and Philip A Romero. Data-driven protein engineering. *Protein Engineering: Tools and Applications*, pages 133–151, 2021.

[65] Nicolas Guex and Manuel C. Peitsch. Swiss-model and the swiss-pdb viewer: An environment for comparative protein modeling. *ELECTROPHORESIS*, 18(15):2714–2723, Jan 1997.

[66] Barry G Hall. Building phylogenetic trees from molecular data with mega. *Molecular biology and evolution*, 30(5):1229–1235, 2013.

[67] Jiawei Han and Jing Gao. Research challenges for data mining in science and engineering. *Next Generation of Data Mining*, pages 1–18, 2009.

[68] Xi Han, Xiaonan Wang, and Kang Zhou. Develop machine learning-based regression predictive models for engineering protein solubility. *Bioinformatics*, 35(22):4640–4646, 2019.

[69] David J Hand. Data mining. *Encyclopedia of Environmetrics*, 2, 2006.

[70] Simon S. Haykin. *Neural networks and learning machines*. Pearson Education, Upper Saddle River, NJ, third edition, 2009.

[71] J Matthew Helm, Andrew M Swiergosz, Heather S Haeberle, Jaret M Karnuta, Jonathan L Schaffer, Viktor E Krebs, Andrew I Spitzer, and Prem N Ramkumar. Machine learning and artificial intelligence: definitions, applications, and future directions. *Current reviews in musculoskeletal medicine*, 13(1):69–76, 2020.

[72] Jesús Herrera-Bravo, Lisandra Herrera Belén, Jorge G Farias, and Jorge F Beltrán. Tap 1.0: A robust immunoinformatic tool for the prediction of tumor t-cell antigens based on aaindex properties. *Computational Biology and Chemistry*, 91:107452, 2021.

[73] Thomas A Hopf, Anna G Green, Benjamin Schubert, Sophia Mersmann, Charlotta PI Schärfe, John B Ingraham, Agnes Toth-Petroczy, Kelly Brock, Adam J Riesselman, Perry Palmedo, et al. The evcouplings python framework for coevolutionary sequence analysis. *Bioinformatics*, 35(9):1582–1584, 2019.

[74] Anil K Jain, Richard C Dubes, et al. *Algorithms for clustering data*, volume 6. Prentice hall Englewood Cliffs, 1988.

[75] Arian R Jamasb, Ben Day, Cătălina Cangea, Pietro Liò, and Tom L Blundell. Deep learning for protein–protein interaction site prediction. In *Proteomics Data Analysis*, pages 263–288. Springer, 2021.

[76] Arian Rokkum Jamasb, Pietro Lió, and Tom Blundell. Graphein-a python library for geometric deep learning and network analysis on protein structures. *bioRxiv*, 2020.

[77] Benjamin T James, Brian B Luczak, and Hani Z Girgis. Meshclust: an intelligent tool for clustering dna sequences. *Nucleic acids research*, 46(14):e83–e83, 2018.

[78] Tsai-Yang Jea and Venu Govindaraju. A minutia-based partial fingerprint recognition system. *Pattern recognition*, 38(10):1672–1684, 2005.

[79] H.Joel Jeffrey. Chaos game representation of gene structure. 18(8):2163–2170, 1990.

[80] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[81] Kazutaka Katoh, Kei-ichi Kuma, Hiroyuki Toh, and Takashi Miyata. Mafft version 5: improvement in accuracy of multiple sequence alignment. *Nucleic acids research*, 33(2):511–518, 2005.

[82] Shuichi Kawashima and Minoru Kanehisa. Aaindex: Amino acid index database. *Nucleic Acids Research*, 28(1):374–374, Jan 2000.

[83] W James Kent. Blat—the blast-like alignment tool. *Genome research*, 12(4):656–664, 2002.

[84] Sofia Khan and Mauno Vihinen. Performance of protein stability predictors. *Human Mutation*, 31(6):675–684, Jun 2010.

[85] Donghwa Kim, Deokseong Seo, Suhyoun Cho, and Pilsung Kang. Multi-co-training for document classification using various document representations: Tf–idf, lda, and doc2vec. *Information Sciences*, 477:15–29, 2019.

[86] D Kolba and TW Parks. A prime factor fft algorithm using high-speed convolution. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(4):281–294, 1977.

[87] Greg Landrum. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling, 2013.

[88] Pedro Larrañaga, Borja Calvo, Roberto Santana, Concha Bielza, Josu Galdiano, Iñaki Inza, José A. Lozano, Rubén Armañanzas, Guzmán Santafé, Aritz Pérez, and Victor Robles. Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7(1):86–112, Mar 2006.

[89] Nguyen Quoc Khanh Le, Edward Kien Yee Yapp, Yu-Yen Ou, and Hui-Yuan Yeh. imotor-cnn: Identifying molecular functions of cytoskeleton motor proteins using 2d convolutional neural network via chou's 5-step rule. *Analytical Biochemistry*, 575:17–26, 2019.

[90] Andrew Leaver-Fay, Michael Tyka, Steven M Lewis, Oliver F Lange, James Thompson, Ron Jacak, Kristian W Kaufman, P Douglas Renfrew, Colin A Smith, Will Sheffler, et al. Rosetta3: an object-oriented software suite for the simulation and design of macromolecules. In *Methods in enzymology*, volume 487, pages 545–574. Elsevier, 2011.

[91] Jae K. Lee, Paul D. Williams, and Sooyoung Cheon. Data mining in genomics. *Clinics in Laboratory Medicine*, 28(1):145–166, 2008.

[92] M. Li, J. Wang, and J. Chen. A fast agglomerate algorithm for mining functional modules in protein interaction networks. In *2008 International Conference on BioMedical Engineering and Informatics*, volume 1, pages 3–7, May 2008.

[93] Min Li, Jianxin Wang, and Jian'er Chen. A fast agglomerate algorithm for mining functional modules in protein interaction networks. In *2008 International conference on biomedical engineering and informatics*, volume 1, pages 3–7. IEEE, 2008.

[94] Yougen Li, D Allan Drummond, Andrew M Sawayama, Christopher D Snow, Jesse D Bloom, and Frances H Arnold. A diverse family of thermostable cytochrome p450s created by recombination of stabilizing fragments. *Nature biotechnology*, 25(9):1051–1056, 2007.

[95] Kathy Liszewski. Speeding up the protein assembly line. *Genetic Engineering & Biotechnology News*, 35(04):1–10, 2015.

[96] Bingchen Liu, Yizhe Zhu, Zuohui Fu, Gerard De Melo, and Ahmed Elgammal. Oogan: Disentangling gan with one-hot sampling and orthogonal regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4836–4843, 2020.

[97] Yunan Luo, Guangde Jiang, Tianhao Yu, Yang Liu, Lam Vo, Hantian Ding, Yufeng Su, Wesley Wei Qian, Huimin Zhao, and Jian Peng. Ecnet is an evolutionary context-integrated deep learning framework for protein engineering. *Nature communications*, 12(1):1–14, 2021.

[98] Yunan Luo, Lam Vo, Hantian Ding, Yufeng Su, Yang Liu, Wesley Wei Qian, Huimin Zhao, and Jian Peng. Evolutionary context-integrated deep sequence modeling for protein engineering. *bioRxiv*, 2020.

[99] Stefan Lutz. Beyond directed evolution—semi-rational protein engineering and design. *Current opinion in biotechnology*, 21(6):734–743, 2010.

[100] Sergey Lyskov and Jeffrey J Gray. The rosettadock server for local protein–protein docking. *Nucleic acids research*, 36(suppl_2):W233–W238, 2008.

[101] Chunwei Ma, Yan Ren, Jiarui Yang, Zhe Ren, Huanming Yang, and Siqi Liu. Improved peptide retention time prediction in liquid chromatography through deep learning. *Analytical chemistry*, 90(18):10881–10888, 2018.

[102] Davide Maltoni, Dario Maio, Anil K Jain, and Salil Prabhakar. *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

[103] Derek M Mason, Simon Friedensohn, Cédric R Weber, Christian Jordi, Bastian Wagner, Simon Meng, Pablo Gainza, Bruno E Correia, and Sai T Reddy. Deep learning enables therapeutic antibody optimization in mammalian cells by deciphering high-dimensional protein sequence space. *BioRxiv*, 2019.

[104] Majid Masso and Iosif I Vaisman. Auto-mute 2.0: a portable framework with enhanced capabilities for predicting protein functional consequences upon mutation. *Advances in bioinformatics*, 2014, 2014.

[105] Stanislav Mazurenko, Zbynek Prokop, and Jiri Damborsky. Machine learning in enzyme engineering. *ACS Catalysis*, 10(2):1210–1223, 2019.

[106] Liam J McGuffin, Jennifer D Atkins, Bajuna R Salehe, Ahmad N Shuid, and Daniel B Roche. Intfold: an integrated server for modelling protein structures and functions from amino acid sequences. *Nucleic acids research*, 43(W1):W169–W173, 2015.

[107] David Medina-Ortiz, Sebastian Contreras, Juan Amado-Hinojosa, Jorge Torres-Almonacid, Juan A Asenjo, Marcelo Navarrete, and Alvaro Olivera-Nappa. Combination of digital signal processing and assembled predictive models facilitates the rational design of proteins. *arXiv preprint arXiv:2010.03516*, 2020.

[108] David Medina-Ortiz, Sebastián Contreras, Cristofer Quiroz, Juan A Asenjo, and Álvaro Olivera-Nappa. Dmakit: A user-friendly web platform for bringing state-of-the-art data analysis techniques to non-specific users. *Information systems*, 93:101557, 2020.

[109] David Medina-Ortiz, Sebastián Contreras, Cristofer Quiroz, and Álvaro Olivera-Nappa. Development of supervised learning predictive models for highly non-linear biological, biomedical, and general datasets. *Frontiers in molecular biosciences*, 7:13, 2020.

[110] Donald Michie, David J Spiegelhalter, CC Taylor, et al. Machine learning. *Neural and Statistical Classification*, 13, 1994.

[111] Mary S Morrison, Christopher J Podracky, and David R Liu. The developing toolkit of continuous directed evolution. *Nature chemical biology*, 16(6):610–619, 2020.

[112] Bertrand Muquet, Zhengdao Wang, Georgios B Giannakis, Marc De Courville, and Pierre Duhamel. Cyclic prefixing or zero padding for wireless multicarrier transmissions? *IEEE Transactions on communications*, 50(12):2136–2148, 2002.

[113] Fionn Murtagh and Pierre Legendre. Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of classification*, 31(3):274–295, 2014.

[114] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[115] Alexandros Nikitas, Kalliopi Michalakopoulou, Eric Tchouamou Njoya, and Dimitris Karampatzakis. Artificial intelligence, transport and the smart city: Definitions and dimensions of a new mobility era. *Sustainability*, 12(7):2789, 2020.

[116] M. Novič and M. Randić†. Representation of proteins as walks in 20-d space. *SAR and QSAR in Environmental Research*, 19(3-4):317–337, 2008. PMID: 18484501.

[117] Mary K. Obenshain. Application of data mining techniques to healthcare data. *Infection Control &; Hospital Epidemiology*, 25(8):690–695, 2004.

[118] Alvaro Olivera-Nappa, Barbara A. Andrews, and Juan A. Asenjo. Mutagenesis objective search and selection tool (mosst): an algorithm to predict structure-function related mutations in proteins. *BMC Bioinformatics*, 12(1):122, Apr 2011.

[119] Margarita Osadchy and Rachel Kolodny. How deep learning tools can help protein engineers find good sequences. *The Journal of Physical Chemistry B*, 125(24):6440–6450, 2021.

[120] Ozlem Ozbudak and Zümray Dokur. Protein fold classification using kohonen's self-organizing map. In *IWBBIO*, pages 903–911, 2014.

[121] Arun Prasad Pandurangan, Bernardo Ochoa-Montaño, David B Ascher, and Tom L Blundell. Sdm: a server for predicting effects of mutations on protein stability. *Nucleic acids research*, 45(W1):W229–W235, 2017.

[122] Arun Prasad Pandurangan, Bernardo Ochoa-Montaño, David B. Ascher, and Tom L. Blundell. Sdm: a server for predicting effects of mutations on protein stability. *Nucleic Acids Res*, 45(W1):W229–W235, Jul 2017. 3835311[PII].

[123] Maria Papagianni. Ribosomally synthesized peptides with antimicrobial properties: biosynthesis, structure, function, and applications. *Biotechnology advances*, 21(6):465–499, 2003.

[124] Antonin Pavelka, Eva Chovancova, and Jiri Damborsky. Hotspot wizard: a web server for identification of hot spots in protein engineering. *Nucleic acids research*, 37(suppl_2):W376–W383, 2009.

[125] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[126] Marharyta Petukh, Luogeng Dai, and Emil Alexov. Saambe: Webserver to predict the charge of binding free energy caused by amino acids mutations. *International Journal of Molecular Sciences*, 17(4), 2016.

[127] James C Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D Skeel, Laxmikant Kale, and Klaus Schulten. Scalable molecular dynamics with namd. *Journal of computational chemistry*, 26(16):1781–1802, 2005.

[128] Vladimir Potapov, Mati Cohen, and Gideon Schreiber. Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details. *Protein Engineering, Design and Selection*, 22(9):553–560, 06 2009.

[129] Kaiyang Qu, Leyi Wei, and Quan Zou. A review of dna-binding proteins prediction methods. *Current Bioinformatics*, 14(3):246–254, 2019.

[130] Cristofer Quiroz, Yasna Barrera Saavedra, Benjamín Armijo-Galdames, Juan Amado-Hinojosa, Álvaro Olivera-Nappa, Anamaria Sanchez-Daza, and David Medina-Ortiz. Peptipedia: a user-friendly web application and a comprehensive database for peptide research supported by machine learning approach. *Database*, 2021, 2021.

[131] M Saifur Rahman, Swakkhar Shatabda, Sanjay Saha, Mohammad Kaykobad, and M Sohel Rahman. Dpp-pseaac: A dna-binding protein prediction model using chou's general pseaac. *Journal of theoretical biology*, 452:22–34, 2018.

[132] Akanksha Rajput, Amit Kumar Gupta, and Manoj Kumar. Prediction and analysis of quorum sensing peptides based on sequence features. *PLoS One*, 10(3):e0120066, 2015.

[133] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole von Lilienfeld. Big data meets quantum chemistry approximations: the δ-machine learning approach. *Journal of chemical theory and computation*, 11(5):2087–2096, 2015.

[134] K Ramamohan Rao and Ping Yip. *Discrete cosine transform: algorithms, advantages, applications*. Academic press, 2014.

[135] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S Song. Evaluating protein transfer learning with tape. In *Advances in Neural Information Processing Systems*, 2019.

[136] M. Rebhan, V. Chalifa-Caspi, J. Prilusky, and D. Lancet. Genecards: a novel functional genomics compendium with automated data mining and query reformulation support. *Bioinformatics*, 14(8):656–664, Sep 1998.

[137] Jonas Reeb, Theresa Wirth, and Burkhard Rost. Variant effect predictions capture some aspects of deep mutational scanning experiments. *BMC Bioinformatics*, 21(1):107, Mar 2020.

[138] Manfred T. Reetz, Daniel Kahakeaw, and Renate Lohmer. Addressing the numbers problem in directed evolution. *ChemBioChem*, 9(11):1797–1804, Jul 2008.

[139] Radim Řehřek, Petr Sojka, et al. Gensim—statistical semantics in python. *Retrieved from genism. org*, 2011.

[140] Schrödinger Release. 1: Maestro. *Schrödinger, LLC, New York, NY, USA*, 2016.

[141] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021.

[142] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[143] David E. Rumelhart, Richard Durbin, Richard Golden, and Yves Chauvin. *Backpropagation: The Basic Theory*, page 1–34. L. Erlbaum Associates Inc., USA, 1995.

[144] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249, 2018.

[145] Samia Tasnim Sara, Md Mehedi Hasan, Ahsan Ahmad, and Swakkhar Shatabda. Convolutional neural networks with image representation of amino acid sequences for protein function prediction. *Computational Biology and Chemistry*, 92:107494, 2021.

[146] Samia Tasnim Sara, Md Mehedi Hasan, Ahsan Ahmad, and Swakkhar Shatabda. Convolutional neural networks with image representation of amino acid sequences for protein function prediction. *Computational Biology and Chemistry*, 92:107494, 2021.

[147] Ajay Kumar Saw, Garima Raj, Manashi Das, Narayan Chandra Talukdar, Binod Chandra Tripathy, and Soumyadeep Nandi. Alignment-free method for dna sequence clustering using fuzzy integral similarity. *Scientific reports*, 9(1):1–18, 2019.

[148] Joost Schymkowitz, Jesper Borg, Francois Stricher, Robby Nys, Frederic Rousseau, and Luis Serrano. The foldx web server: an online force field. *Nucleic acids research*, 33(suppl_2):W382–W388, 2005.

[149] Joost Schymkowitz, Jesper Borg, Francois Stricher, Robby Nys, Frederic Rousseau, and Luis Serrano. The foldx web server: an online force field. *Nucleic Acids Res*, 33(Web Server issue):W382–W388, Jul 2005.

[150] Stephen B Seidman and Brian L Foster. A graph-theoretic generalization of the clique concept. *Journal of Mathematical sociology*, 6(1):139–154, 1978.

[151] Mark Sendak, Madeleine Clare Elish, Michael Gao, Joseph Futoma, William Ratliff, Marshall Nichols, Armando Bedoya, Suresh Balu, and Cara O'Brien. "the human body is a black box"supporting clinical decision-making with deep learning. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 99–109, 2020.

[152] Yin Xiang Setoh, Alberto A. Amarilla, Nias Y. G. Peng, Rebecca E. Griffiths, Julio Carrera, Morgan E. Freney, Eri Nakayama, Shinya Ogawa, Daniel Watterson, Naphak Modhiran, Faith Elizabeth Nanyonga, Francisco J. Torres, Andrii Slonchak, Parthiban Periasamy, Natalie A. Prow, Bing Tang, Jessica Harrison, Jody Hobson-Peters, Thom Cuddihy, Justin Cooper-White, Roy A. Hall, Paul R. Young, Jason M. Mackenzie, Ernst Wolvetang, Jesse D. Bloom, Andreas Suhrbier, and Alexander A. Khromykh. Determinants of zika virus host tropism uncovered by deep mutational scanning. *Nature Microbiology*, 4(5):876–887, May 2019.

[153] HyeonSeok Shin and Byung-Kwan Cho. Rational protein engineering guided by deep mutational scanning. *International journal of molecular sciences*, 16(9):23094–23110, 2015.

[154] Ajay Shrestha and Ausif Mahmood. Review of deep learning algorithms and architectures. *IEEE Access*, 7:53040–53065, 2019.

[155] Raghav Shroff, Austin W Cole, Barrett R Morrow, Daniel J Diaz, Isaac Donnell, Jimmy Gollihar, Andrew D Ellington, and Ross Thyer. A structure-based deep learning framework for protein engineering. *bioRxiv*, page 833905, 2019.

[156] Niklas E Siedhoff, Alexander-Maurice Illig, Ulrich Schwaneberg, and Mehdi D Davari. Pypef—an integrated framework for data-driven protein engineering. *Journal of Chemical Information and Modeling*, 61(7):3463–3476, 2021.

[157] Niklas E Siedhoff, Ulrich Schwaneberg, and Mehdi D Davari. Machine learning-assisted enzyme engineering. *Methods in Enzymology*, 643:281–315, 2020.

[158] Narasimhaiah Sitaram and Ramakrishnan Nagaraj. Host-defense antimicrobial peptides: importance of structure for activity. *Current pharmaceutical design*, 8(9):727–742, 2002.

[159] Ian Naismith Sneddon. *Fourier transforms*. Courier Corporation, 1995.

[160] X-D Sun and R-B Huang. Prediction of protein structural classes using support vector machines. *Amino acids*, 30(4):469–475, 2006.

[161] Z. Sun, S. Pei, R. L. He, and S. S. Yau. A novel numerical representation for proteins: Three-dimensional Chaos Game Representation and its Extended Natural Vector. *Comput Struct Biotechnol J*, 18:1904–1913, 2020.

[162] Yuefeng Tang, Yuanpeng Janet Huang, Thomas A Hopf, Chris Sander, Debora S Marks, and Gaetano T Montelione. Protein structure determination by combining sparse nmr data with evolutionary couplings. *Nature methods*, 12(8):751–754, 2015.

[163] Julie D. Thompson, Toby J. Gibson, Frédéric Plewniak, François Jeanmougin, and Desmond G. Higgins. The clustal_x windows interface: Flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucleic Acids Research*, 25(24):4876–4882, Dec 1997.

[164] Andrey Tovchigrechko and Ilya A Vakser. Gramm-x public web server for protein–protein docking. *Nucleic acids research*, 34(suppl_2):W310–W314, 2006.

[165] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[166] Salman Sadullah Usmani, Sherry Bhalla, and Gajendra PS Raghava. Prediction of antitubercular peptides from sequence information using ensemble classifier and hybrid features. *Frontiers in pharmacology*, 9:954, 2018.

[167] Rocio Vargas, Amir Mosavi, and Ramon Ruiz. Deep learning: A review. *Preprints*, 05 2017.

[168] V Veljkovic, I Cosic, D Lalovic, et al. Is it possible to analyze dna and protein sequences by the methods of digital signal processing? *IEEE Transactions on Biomedical Engineering*, (5):337–341, 1985.

[169] Sheng Wang, Zhen Li, Yizhou Yu, and Jinbo Xu. Folding membrane proteins by deep transfer learning. *Cell systems*, 5(3):202–211, 2017.

[170] Leyi Wei, Jijun Tang, and Quan Zou. Local-dpp: An improved dna-binding protein prediction method by exploring local evolutionary information. *Information Sciences*, 384:135 – 144, 2017.

[171] Jochen Weile, Song Sun, Atina G Cote, Jennifer Knapp, Marta Verby, Joseph C Mellor, Yingzhou Wu, Carles Pons, Cassandra Wong, Natascha van Lieshout, Fan Yang, Murat Tasan, Guihong Tan, Shan Yang, Douglas M Fowler, Robert Nussbaum, Jesse D Bloom, Marc Vidal, David E Hill, Patrick Aloy, and Frederick P Roth. A framework for exhaustively mapping functional missense variants. *Molecular Systems Biology*, 13(12):957, 2017.

[172] Peter Welch. The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15(2):70–73, 1967.

[173] Greg Winter. Synthetic human antibodies and a strategy for protein engineering. *FEBS letters*, 430(1-2):92–94, 1998.

[174] Bruce J Wittmann, Kadina E Johnston, Zachary Wu, and Frances H Arnold. Advances in machine learning for directed evolution. *Current Opinion in Structural Biology*, 69:11–18, 2021.

[175] Catherine L Worth, Robert Preissner, and Tom L Blundell. Sdm—a server for predicting effects of mutations on protein stability and malfunction. *Nucleic acids research*, 39(suppl_2):W215–W222, 2011.

[176] Zachary Wu, Kadina E Johnston, Frances H Arnold, and Kevin K Yang. Protein sequence design with deep generative models. *Current opinion in chemical biology*, 65:18–27, 2021.

[177] Zachary Wu, SB Jennifer Kan, Russell D Lewis, Bruce J Wittmann, and Frances H Arnold. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences*, 116(18):8852–8858, 2019.

[178] Xuan Xiao, Pu Wang, Wei-Zhong Lin, Jian-Hua Jia, and Kuo-Chen Chou. iamp-2l: a two-level multi-label classifier for identifying antimicrobial peptides and their functional types. *Analytical biochemistry*, 436(2):168–177, 2013.

[179] Yuting Xu, Deeptak Verma, Robert P Sheridan, Andy Liaw, Junshui Ma, Nicholas M Marshall, John McIntosh, Edward C Sherer, Vladimir Svetnik, and Jennifer M Johnston. Deep dive into machine learning models for protein engineering. *Journal of chemical information and modeling*, 60(6):2773–2790, 2020.

[180] Hui Yang, Srinivasan Parthasarathy, and Sameep Mehta. A generalized framework for mining spatio-temporal patterns in scientific data. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 716–721, New York, NY, USA, 2005. ACM.

[181] Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature methods*, 16(8):687–694, 2019.

[182] Kevin K Yang, Zachary Wu, Claire N Bedbrook, and Frances H Arnold. Learned protein embeddings for machine learning. *Bioinformatics*, 34(15):2642–2648, 2018.

[183] Hai-Cheng Yi, Zhu-Hong You, Xi Zhou, Li Cheng, Xiao Li, Tong-Hai Jiang, and Zhan-Heng Chen. Acp-dl: a deep learning long short-term memory model to predict anticancer peptides using high-efficiency feature representation. *Molecular Therapy-Nucleic Acids*, 17:1–9, 2019.

[184] Illhoi Yoo, Patricia Alafaireet, Miroslav Marinov, Keila Pena-Hernandez, Rajitha Gopidi, Jia-Fu Chang, and Lei Hua. Data mining in healthcare and biomedicine: A survey of the literature. *Journal of Medical Systems*, 36(4):2431–2448, Aug 2012.

[185] Zhu-Hong You, Jianqiang Li, Jianqiang Li, Xin Gao, Xin Gao, Xin Gao, Xin Gao, Zhou He, Lin Zhu, Lin Zhu, Ying-Ke Lei, Ying-Ke Lei, and Zhiwei Ji. Detecting protein-protein interactions with a novel matrix-based protein sequence representation and support vector machines. *BioMed Research International*, 2015.

[186] Jia-Feng Yu, Xianghua Dou, Hong-Bo Wang, Xiao Sun, Huiying Zhao, and Ji-Hua Wang. A novel cylindrical representation for characterizing intrinsic properties of protein sequences. *Journal of chemical information and modeling*, 55 6:1261–70, 2015.

[187] Lu Zhang, Jianjun Tan, Dan Han, and Hao Zhu. From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug discovery today*, 22(11):1680–1685, 2017.

[188] David Zimmer, Kevin Schneider, Frederik Sommer, Michael Schroda, and Timo Mühlhaus. Artificial intelligence understands peptide observability and assists with absolute protein quantification. *Frontiers in plant science*, 9:1559, 2018.