Universidad Politécnica
de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Máster Universitario en Inteligencia Artificial

Trabajo Fin de Máster

Predicción de Interacciones Proteína-Proteína basada en Secuencia mediante Modelos de Aprendizaje Profundo

Autor: Óscar Pérez Camacho
Tutor: Alfonso Rodríguez-Patón Aradas

Madrid, julio de 2025

Este Trabajo Fin de Máster se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Máster
Máster Universitario en Inteligencia Artificial

Título: Predicción de Interacciones Proteína-Proteína basada en Secuencia mediante Modelos de Aprendizaje Profundo

julio de 2025

Autor: Óscar Pérez Camacho
Tutor: Alfonso Rodríguez-Patón Aradas
Departamento de Inteligencia Artificial
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

Las proteínas son máquinas moleculares que realizan la mayoría de funciones dinámicas necesarias para la vida. En muchas ocasiones, no trabajan de forma aislada, sino que interactúan entre sí para ejercer su actividad biológica, dando lugar a una compleja red de interacciones proteína-proteína (PPI) en el organismo. El conocimiento de estas redes resulta esencial para entender los procesos celulares, tratar enfermedades y desarrollar fármacos. Sin embargo, los métodos experimentales disponibles presentan limitaciones en precisión, tiempo y coste, que restringen el descubrimiento masivo de PPI. Debido a esto, la predicción computacional es una opción atractiva para promover el avance en líneas de investigación y guiar el descubrimiento experimental. La predicción basada únicamente en la secuencia de las proteínas resulta especialmente interesante, ya que es la información más ampliamente disponible.

En los últimos años, el aprendizaje profundo ha impulsado avances extraordinarios en el análisis de proteínas, permitiendo establecer relaciones entre su secuencia, estructura y función que resultan muy costosas de determinar de forma experimental. Por ello, el aprendizaje profundo resulta una estrategia prometedora para la predicción de PPI basada en secuencia. Sin embargo, los métodos diseñados hasta el momento presentan rendimientos sobreestimados debido al uso de datasets con filtraciones de datos entre el conjunto de entrenamiento y validación. En ausencia de filtraciones, sus predicciones se vuelven azarosas, destacando la necesidad de continuar la investigación en este área, poniendo especial cuidado en el diseño de los conjuntos de datos.

En este trabajo se han explorado ProtBERT, un modelo de lenguaje de proteínas, y AlphaFold3, el estado del arte en el modelado de estructura de complejos proteicos, para la predicción de PPI a partir de secuencia en ausencia de filtraciones de datos. Los resultados muestran que las representaciones internas de AlphaFold3 codifican información útil para la clasificación de pares de proteínas en interactores o no interactores. Una regresión logística aplicada sobre las representaciones individuales extraídas por AlphaFold3, fue suficiente para predecir interacciones proteína-proteína con gran precisión y sensibilidad (accuracy 0.840, precisión 0.842, sensibilidad 0.836). En contraposición, ProtBERT fue incapaz de extraer características relevantes. Los resultados muestran la importancia de utilizar modelos capaces de extraer información estructural a partir de la secuencia, como AlphaFold3, para lograr una predicción eficaz de las interacciones. Este trabajo inicia el camino para adaptar AlphaFold3 y otros modelos especializados en la generación de estructuras de complejos multiméricos, a la tarea de predicción de interacciones proteína-proteína.

Abstract

Proteins are molecular machines that perform most of the dynamic functions necessary for life. In many cases, they do not work in isolation. Instead, they interact with each other to exert their biological activity, giving rise to a complex network of protein-protein interactions (PPI) in the organism. Knowledge of these networks is essential for understanding cellular processes, treating diseases and developing drugs. However, available experimental methods present limitations in accuracy, time and cost, which restrict the mass discovery of PPI. Because of this, computational prediction is an attractive option to promote progress in research and guide experimental discovery. Predictions based on protein sequence alone are particularly interesting, as this is the most widely available information.

In recent years, deep learning has driven extraordinary advances in protein analysis, making it possible to establish relationships between sequence, structure and function that are very costly to determine experimentally. Thus, deep learning is a promising strategy for sequence-based PPI prediction. However, the methods designed so far present overestimated performances due to the use of datasets with data leaks between training and validation. In absence of data leakage, their predictions become random, underpinning the necessity of continuing research in this field, paying especial attention to the design of datasets.

In this work ProtBERT, a protein language model, and AlphaFold3, the state of the art in protein complex structure modeling, have been explored for PPI prediction from sequence in absence of data leakage. The results show that the internal representations of AlphaFold3 encode useful information for the classification of protein pairs into interacting or non-interacting. A logistic regression applied on the individual representations extracted by AlphaFold3 was sufficient to predict protein-protein interactions with high precision and sensitivity (accuracy 0.840, precision 0.842, sensitivity 0.836). In contrast, ProtBERT was unable to extract relevant features. The results show the importance of using models capable of extracting structural information from sequence, such as AlphaFold3, for efficient prediction of interactions. This work initiates the path towards adapting AlphaFold3 and other sequence-to-structure models of protein complexes for PPI prediction.

Tabla de contenidos

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Las proteínas, su estructura e interacciones

Las proteínas son macromoléculas complejas formadas por cadenas de aminoácidos, capaces de desempeñar una gran diversidad de funciones que las hacen esenciales para los seres vivos (Campbell *et al.*, 2021). Algunas tienen función estructural, como el colágeno que mantiene la matriz extracelular de los animales. Otras actúan como catalizadores, reduciendo la energía de activación necesaria para que las reacciones químicas del metabolismo ocurran a una velocidad compatible con la vida. También existen proteínas con función reguladora, que ajustan el metabolismo según las necesidades del organismo para mantener la homeostasis. Ejemplos de esto son las hormonas proteicas, como la insulina, que permiten la comunicación entre células, o los factores de transcripción, que controlan la expresión génica dentro de la célula. Además, algunas proteínas participan en el transporte o almacenamiento de sustancias, como la transferrina o la hemoglobina, que transportan hierro y oxígeno en la sangre respectivamente. Finalmente, hay proteínas que presentan funciones especializadas. Los anticuerpos intervienen en la defensa del organismo. Las proteínas que conforman las miofibrillas musculares, cilios y flagelos, permiten el movimiento a diferentes escalas. Algunas bacterias patógenas, producen proteínas llamadas efectores, que secretan en el interior de la célula huésped, para modificar su funcionamiento normal y facilitar la infección.

Las proteínas se caracterizan por su secuencia y estructura tridimensional. En condiciones nativas, adoptan un plegamiento tridimensional de mínima energía, que resulta esencial para desempeñar su función (Campbell *et al.*, 2021). En concreto, las proteínas presentan una estructura jerárquica compleja, definida en cuatro niveles: estructura primaria, secundaria, terciaria y cuaternaria (figura 1.1). La estructura primaria alude a la secuencia lineal de aminoácidos (también llamados residuos, porque se encuentran polimerizados) que forman la cadena. Los aminoácidos son moléculas orgánicas pequeñas que presentan un grupo amino (-NH_3) y carboxilo (-COOH) sobre el mismo carbono, llamado carbono Cα, al cual también se une una cadena lateral (-R). Dependiendo de la cadena lateral, existen 20 aminoácidos distintos encargados de formar las proteínas en los seres vivos. Estos pueden polimerizar en cualquier orden dando lugar a una cadena polipeptídica, es decir, una proteína.

La estructura primaria determina la estructura química de la proteína y por tanto

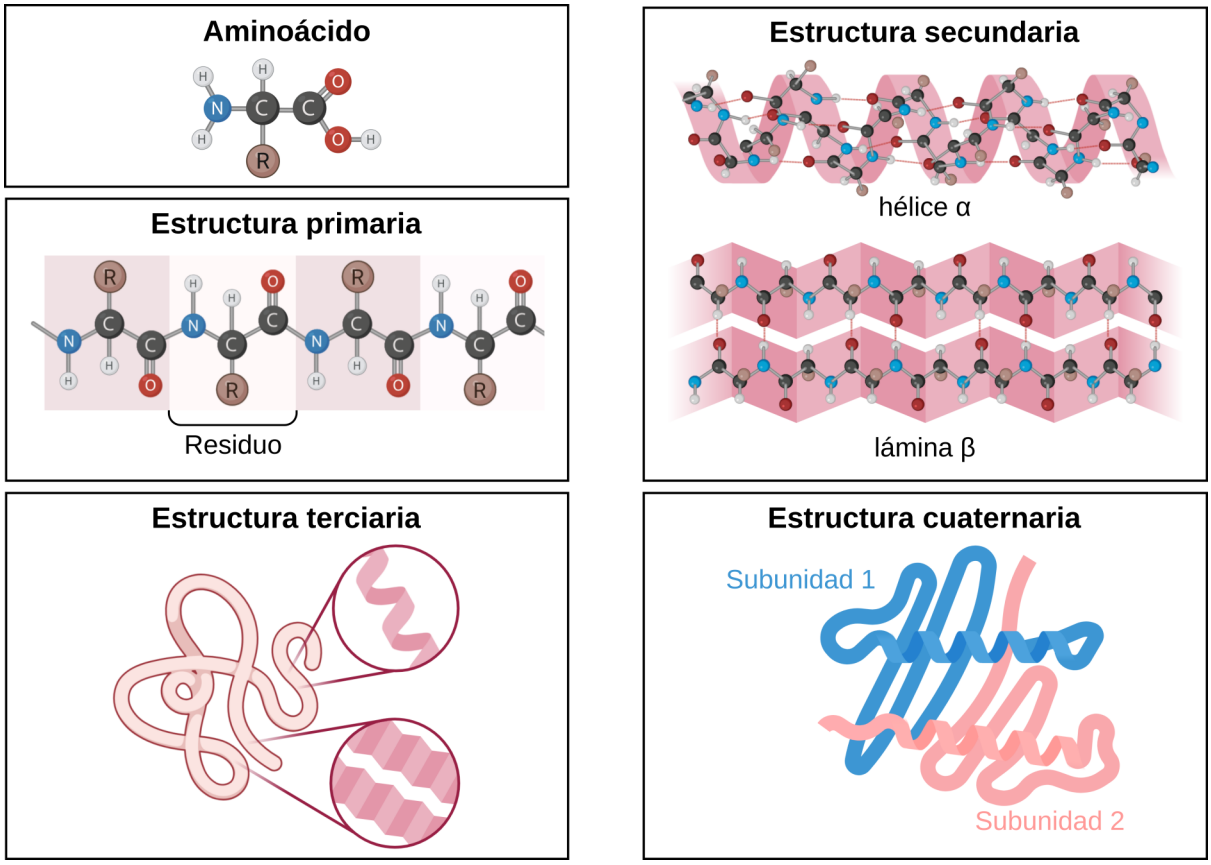

Figura 1.1: Estructura jerárquica de las proteínas.

es el principal condicionante de su plegamiento tridimensional. Este plegamiento se produce a dos niveles: estructura secundaria y terciaria. La estructura secundaria consiste en estructuras regulares como las hélices α y láminas β, debidas al establecimiento de interacciones intramoleculares locales de enlaces de hidrógeno entre los grupos amino y carbonilo de los residuos. La estructura terciaria es la forma tridimensional de la proteína, resultado de interacciones intramoleculares entre las cadenas laterales de los residuos. Finalmente, algunas proteínas son capaces de formar complejos multiméricos estables con otras. Las distintas subunidades proteicas interactúan entre sí y se organizan en el espacio dando lugar a una estructura cuaternaria.

En muchas ocasiones, las proteínas no ejercen su función de manera aislada, sino que su actividad biológica depende de su capacidad para interactuar con otras proteínas. Las interacciones proteína-proteína (PPI) son fundamentales para una amplia variedad de procesos celulares. Por ejemplo, la insulina debe unirse a su receptor de membrana –otra proteína localizada en la superficie celular– para ser reconocida y desencadenar una compleja cascada de señalización, mediada por sucesivas interacciones entre proteínas reguladoras, que culmina en efectos fisiológicos como la entrada de glucosa al interior celular (Virkamäki *et al.*, 1999). Asimismo, los anticuerpos actúan uniéndose específicamente a epítopos presentes en proteínas antigénicas de patógenos, lo cual señaliza su presencia a los leucocitos y facilita su eliminación (Ramaraj *et al.*, 2012). Por otro lado, las fibras de actina, un elemento estructural fundamental de las miofibrillas musculares y el citoesqueleto de eucariotas, se for-

man a través de la interacción y polimerización de subunidades globulares de actina (Cooper, 2000).

Las PPI se pueden clasificar según sus características estructurales y cinéticas (Akbarzadeh *et al.*, 2024). Pueden ser directas, cuando dos proteínas se acoplan físicamente mediante interacciones intermoleculares, o indirectas, cuando no están en contacto físico, pero interactúan a través de otras proteínas que forman parte del mismo complejo. Además, según su cinética, las PPI pueden ser permanentes o transitorias. Las interacciones permanentes permiten formar complejos estables, mientras que las transitorias son interacciones dinámicas útiles en procesos biológicos como la señalización celular o la regulación génica.

En adelante, este trabajo se centrará en las interacciones directas, independientemente de su cinética. Estas interacciones son deterministas y específicas, es decir, su ocurrencia no es azarosa y está determinada por la secuencia y estructura de las proteínas implicadas. Estas interacciones específicas requieren una complementariedad de forma en la superficie de contacto (interfaz de interacción), así como el establecimiento de fuerzas intermoleculares que las estabilicen: interacciones hidrofóbicas, enlaces de hidrógeno, interacciones electrostáticas y fuerzas de Van der Waals (Lawrence & Colman, 1993; Seychell & Beck, 2021).

Las PPI representan un importante objeto de investigación, no solo por su papel central en numerosos procesos celulares, sino también por su relevancia en mecanismos patogénicos. Muchas bacterias patógenas presentan efectores, que son inyectados en el interior de las células del huésped mediante sistemas de secreción, habitualmente tipo III (T3SS), tipo IV (T4SS) y tipo VI (T6SS) (Schroeder *et al.*, 2021). Estos efectores suelen ser factores críticos para la virulencia de las bacterias, lo cual los convierte en interesantes dianas terapéuticas. Los efectores manipulan las rutas celulares a través del establecimiento de PPI con las proteínas del huésped. Mediante la formación de esta nueva red de interacciones, las bacterias patógenas son capaces de modular la actividad celular para beneficio del patógeno, permitiendo su supervivencia y replicación en el huésped (Mocăniță *et al.*, 2025). Por tanto, el estudio de PPI patógeno-huésped resulta esencial para entender el mecanismo de patogenicidad y diseñar fármacos que lo inhiban.

La gran relevancia del descubrimiento de PPI para la comprensión de la actividad proteica, procesos celulares y el desarrollo de fármacos ha incentivado el diseño de métodos experimentales de alta capacidad. Métodos como Yeast 2 Hybrid (Y2H) y la purificación por afinidad acoplada a espectrometría de masas (AP-MS), permiten el descubrimiento de PPI a gran escala (Akbarzadeh *et al.*, 2024). Sin embargo, carecen de una gran precisión, siendo susceptibles a falsos positivos, por lo que requieren una costosa validación posterior. Además, su diseño y realización conlleva un gran tiempo y coste, por lo que hoy en día las redes de PPI experimentales siguen siendo incompletas. Debido a esto, las herramientas de predicción computacionales son atractivas para obtener PPI potenciales, que pueden guiar la búsqueda experimental e incluso ser directamente útiles en investigación. Se han desarrollado numerosas técnicas con distinto grado de fiabilidad. Algunas de ellas utilizan medidas indirectas de la probabilidad de interacción, por ejemplo, la proximidad cromosómica, ontología génica y perfil filogenético (Akbarzadeh *et al.*, 2024). Por otro lado, las técnicas de docking proteína-proteína, como ClusPro (Kozakov *et al.*, 2017) o HDOCK (Yan *et al.*, 2017), utilizan información estructural. Su funcionamiento busca encontrar la es-

tructura del complejo dada la estructura de las proteínas individuales, atendiendo a la complementariedad de forma e interacciones fisicoquímicas que guían la interacción en la realidad. Sin embargo, estas herramientas se enfrentan a problemas como el cambio de conformación estructural entre proteínas individuales y unidas, y principalmente, la baja disponibilidad de información estructural experimental, que las descarta para la predicción en grandes redes de PPI.

En los últimos años, el aprendizaje profundo ha impulsado avances significativos en la biología computacional. Su capacidad para extraer características relevantes de forma automática, lo ha posicionado a la vanguardia del análisis bioinformático de proteínas. Gracias a esta capacidad, la secuencia de una proteína suele ser suficiente para inferir con precisión tanto información estructural como funcional. Por ello, el aprendizaje profundo resulta una estrategia prometedora para la predicción masiva de PPI, y numerosos estudios recientes se han enfocado en su aplicación. A continuación, se describen algunos de los principales avances logrados mediante arquitecturas de aprendizaje profundo en bioinformática, así como el estado del arte en predicción de PPI con estas herramientas.

1.2. Avances relevantes del aprendizaje profundo en bioinformática

1.2.1. Predicción de estructura a partir de secuencia

Como se ha mencionado previamente, la estructura de una proteína está íntimamente ligada con su función y las interacciones que puede formar. Sin embargo, la obtención experimental de la estructura de las proteínas es costosa, haciendo esta información mucho menos disponible que su secuencia. Debido a esto, ha habido un gran interés en diseñar métodos capaces de predecir la estructura de las proteínas a partir de su secuencia. Este desafío se conoce como el problema del plegamiento proteico y ha sido objetivo de investigación durante décadas (Dill *et al.*, 2008). Para incentivar el desarrollo de modelos con este objetivo se creó CASP[1] (Critical Assessment of Techniques for Protein Structure Prediction), una evaluación crítica bienial de los modelos del estado del arte para la predicción de estructura a partir de secuencia (Moult *et al.*, 1995). Desde su primera edición en 1994, se han desarrollado distintos métodos de predicción de estructura: basados en homología, es decir, en similitud de secuencia a otras proteínas con estructura conocida; métodos de reconocimiento de plegamiento, y métodos de predicción ab initio (Fiser, 2010; Lee *et al.*, 2017). Sin embargo, estos métodos seguían presentando grandes problemas en la predicción de proteínas con poca similitud de secuencia a otras con estructura conocida. En CASP14, celebrada en 2020, AlphaFold2 revolucionó el estado del arte en predicción de estructura, consiguiendo una precisión en la reconstrucción de estructuras de proteínas monoméricas similar a la de métodos experimentales (Jumper *et al.*, 2021). Este ha sido uno de los primeros grandes avances en biología molecular impulsado por el aprendizaje profundo, mostrando su gran potencial de aplicación en la modelización de biomoléculas. Desde entonces, otros modelos han conseguido replicar resultados similares, como es el caso de RoseTTAFold (Baek *et al.*, 2021). Aunque se aventura a decir que el problema del plegamiento fue resuelto desde entonces, CASP se sigue desarrollando en búsqueda de avances en tareas más complejas como

[1] https://predictioncenter.org/

la predicción de estructura complejos multiméricos, métricas de confianza, ácidos nucleicos, y complejos proteína-ligando.

1.2.2. Modelos de lenguaje de proteínas

Los grandes modelos de lenguaje (LLM) basados en la arquitectura transformer (Vaswani *et al.*, 2023) han revolucionado el área del procesamiento del lenguaje natural (NLP), dando lugar a avances significativos en generación y comprensión de texto, y denotando una gran capacidad para capturar el contexto y la estructura del lenguaje. Los LLM generativos más recientes, como GPT-4, OpenAI o3 o DeepSeek-R1, demuestran capacidades extraordinarias en tareas complejas como la comprensión lectora, la respuesta a preguntas, e incluso el razonamiento lógico (OpenAI, 2025). El éxito de esta arquitectura en el lenguaje natural ha incentivado su expansión a otras áreas. Tal es el caso de los modelos de lenguaje específicos de proteínas (pLM), que de forma similar están revolucionando el análisis de proteínas, permitiendo establecer relaciones entre secuencia y función que resultan muy costosas de forma experimental. Los pLM se han aplicado satisfactoriamente en una gran variedad de predicciones, como la inferencia de estructura secundaria, regiones intrínsecamente desordenadas, plegamiento, estabilidad y solubilidad (Elnaggar *et al.*, 2022; Lin *et al.*, 2023; Pang & Liu, 2023). Asimismo, han tenido éxito en la predicción de función, actividad enzimática, efecto de mutaciones, diseño de proteínas, etc. (Ferruz & Höcker, 2022; Meier *et al.*, 2021).

De manera similar a los modelos de lenguaje natural, los pLM se entrenan de forma no supervisada en tareas que permiten aprender representaciones generales de las secuencias con información contextual, como la tarea de modelado del lenguaje por enmascaramiento, predicción del siguiente token o permutación de secuencia (Xiao *et al.*, 2025). El entrenamiento de los pLM se realiza utilizando grandes bases de datos de secuencias proteicas, en vez de texto natural. En lugar de «subpalabras», cada aminoácido de la secuencia constituye un token de entrada al modelo, formando un alfabeto de 20 aminoácidos más los tokens especiales. Una vez entrenados, los pLM son capaces de extraer representaciones (*embeddings*) ricas en información sobre las secuencias de entrada, que se pueden trasladar directamente a tareas específicas como la clasificación de secuencia (función, actividad enzimática, solubilidad), clasificación de tokens (regiones desordenadas, estructura secundaria) y regresión de tokens o secuencia para la predicción de propiedades. Los pLM preentrenados se pueden utilizar directamente como extractores de características para estas tareas, con bastante éxito, aunque recientes estudios sugieren que el fine-tuning con una pequeña de cantidad de datos de la tarea específica ofrece mejores resultados (Schmirler *et al.*, 2024).

ESM-1b (Rives *et al.*, 2021), desarrollado por Facebook Research AI, constituye un ejemplo de pLM temprano basado en la arquitectura transformer. En su estudio, entrenaron varios modelos transformer de distinta profundidad formados únicamente por la pila de *encoders*, y LSTM bidireccionales en las casi 250 millones secuencias de proteínas de UniParc, con una cantidad de tokens comparable al tamaño de los grandes datasets de texto utilizados para entrenar LLM. Los modelos transformer mostraron un mejor rendimiento en la tarea de desenmascaramiento que los LSTM con una cantidad similar de parámetros. Asimismo, se determinó empíricamente un aumento del rendimiento de los modelos con el número de parámetros, de forma similar a

las observaciones realizadas en NLP. Las representaciones aprendidas por ESM-1b demostraron contener información biológica a varios niveles, incluyendo propiedades bioquímicas de los aminoácidos, información evolutiva y filogenética, así como información sobre la estructura secundaria y terciaria de las proteínas. Esta información emergió de forma no supervisada simplemente a través de la resolución de la tarea de modelizado de secuencia de proteínas, sin necesidad de realizar un fine-tuning posterior. Los autores atribuyen esta capacidad de extracción de información biológica sin necesidad de conocimiento de dominio, al hecho de que la variabilidad de las secuencias depende de estas variables ocultas, por lo que el modelo debe aprender a representarlas para poder modelizar satisfactoriamente las secuencias de proteínas. La capacidad de este modelo para extraer información biológica ha sido aprovechada posteriormente en la creación de ESMFold (Lin *et al.*, 2023), capaz de predecir estructura de proteínas a partir de su secuencia, y ESM-IF1 (Hsu *et al.*, 2022), capaz de resolver el problema inverso del plegamiento, es decir, predecir la secuencia de una proteína dada su estructura.

En un estudio posterior, varios LLM tradicionalmente usados en NLP fueron trasladados directamente al vocabulario de las proteínas con gran éxito (Elnaggar *et al.*, 2022). Se reentrenaron de forma no supervisada T5, BERT, Albert, TransformerXL y XLNet con mínimos cambios de arquitectura en grandes bases de secuencias como UniRef100 y BFD. Los modelos resultantes (ProtT5, ProtBERT, ProtElectra, etc), al igual que ESM-1b, demostraron generar embeddings con información fisicoquímica, estructural y filogenética sin necesidad de una etapa de entrenamiento supervisada. El uso de estos embeddings en tareas supervisadas de clasificación por residuo y por secuencia, tuvo éxito tras el entrenamiento únicamente de las cabezas de clasificación. Tradicionalmente, muchos modelos de secuencia de proteínas han requerido información evolutiva como entrada, en la forma de alineamientos múltiples de secuencia (MSA). Un ejemplo mencionado previamente es AlphaFold2. Sin embargo, la construcción de los MSA resulta muy costosa computacionalmente. En este sentido, la capacidad de los pLMs de inferir esta información evolutiva sin necesidad de aportarla como entrada les aporta una ventaja en eficiencia computacional.

1.3. Predicción de PPI basada en secuencia

En los últimos años, se han desarrollado una serie de arquitecturas de aprendizaje profundo capaces de predecir si dos proteínas interactúan o no, únicamente a partir de su secuencia. Algunas de las más relevantes en el estado del arte son DeepFE (Yao *et al.*, 2019), PIPR (Chen *et al.*, 2019), D-SCRIPT (Sledzieski *et al.*, 2021) y Topsy-Turvy (Singh *et al.*, 2022). DeepFE utiliza un algoritmo Word2Vec *skip-gram* para aprender una representación de cada aminoácido de la secuencia, seguido de una red neuronal profunda como clasificador. Por otro lado, PIPR emplea una arquitectura siamesa RCNN (red neuronal convolucional recurrente), donde cada proteína se procesa independientemente y los resultados se combinan al final para la clasificación. D-SCRIPT utiliza una arquitectura de aprendizaje profundo para predecir un mapa de contactos (información estructural), a partir del cual se obtiene la clasificación. De esta forma, el modelo pretende realizar una predicción informada en estructura. Finalmente, Topsy-Turvy es un modelo que enriquece la predicción a partir de secuencia de D-SCRIPT con características extraídas de redes de PPI.

La utilidad de estos modelos resulta incuestionable, dado que la información se-

cuencial es ampliamente disponible y la búsqueda experimental de PPI es costosa y requiere mucho tiempo. Sus artículos muestran rendimientos muy elevados, con valores de accuracy superiores a 0.9 en datasets balanceados. Esto invita a pensar que la predicción de PPI en base a secuencia es un problema resuelto en la actualidad. Sin embargo, una publicación reciente ha reabierto el problema, sugiriendo que estos modelos no realizan predicciones en base a propiedades biológicas relevantes. En su lugar, el alto rendimiento puede explicarse completamente debido al uso de datasets con data leakage entre los conjuntos de entrenamiento y validación (Bernett *et al.*, 2024).

Según Bernett *et al.* (2024), existen tres explicaciones para que un clasificador prediga correctamente si dos proteínas $p1$ y $p2$ interaccionan:

- Explicación 1: el modelo es capaz de detectar los patrones biológicos y fisicoquímicos en las secuencias de $p1$ y $p2$ que son responsables de su interacción.
- Explicación 2: el modelo basa su predicción en información de grado presente en las redes de pares de proteínas interactores y no interactores del conjunto de datos de entrenamiento. Si en el entrenamiento la proteína $p1$ aparece únicamente en pares interactores (pares positivos), el modelo tenderá a predecir cualquier par que involucre $p1$ como interactor. De forma similar, si $p1$ aparece mayormente en pares no interactores (pares negativos), el modelo aprenderá este patrón.
- Explicación 3: el modelo realiza su predicción en base a la similitud de secuencia del par $p1$ y $p2$ a otro par $p1'$ y $p2'$ del conjunto de entrenamiento. Esta explicación surge de que proteínas homólogas, con alta similitud de secuencia, suelen conservar sus interacciones.

Las explicaciones 2 y 3 también pueden combinarse, de forma que si una proteína $p1$ es similar a otra proteína $p1'$ que está presente mayoritariamente en pares positivos o negativos del entrenamiento, esto también podría sesgar las predicciones de pares que involucran $p1$.

Los datasets de PPI habitualmente utilizados presentan proteínas compartidas entre los conjuntos de entrenamiento y validación, lo que posibilita la explicación 2. Asimismo, la presencia de proteínas distintas con alta similitud de secuencia entre entrenamiento y validación permite obtener buenos resultados por medio de la explicación 3. Bernett *et al.* (2024) demuestran que las explicaciones 2 y 3 son las principales responsables del excelente rendimiento de los modelos de aprendizaje profundo hasta entonces publicados, y descartan que estos presenten capacidad de detectar los patrones biológicos causantes de la interacción. De esta forma, los modelos no resultan útiles para el descubrimiento de nuevas interacciones, entre proteínas que no se encuentren o sean similares a las del conjunto de entrenamiento. Esto los hace idénticos en rendimiento a modelos de aprendizaje automático más simples que toman como entrada la similitud de secuencia a proteínas del conjunto de entrenamiento y el grado positivo y negativo de las proteínas en entrenamiento, que requieren mucho menor gasto energético y computacional que las arquitecturas profundas. Asimismo, Bernett *et al.* (2024) muestran que SPRINT (Li & Ilie, 2017), un método algorítmico basado únicamente en similitud de secuencia, presenta resultados similares a los modelos de aprendizaje profundo en los datasets de mayor tamaño, demostrando que encontrar subsecuencias similares es suficiente para conseguir rendimientos

excelentes cuando el entrenamiento y validación presentan proteínas compartidas.

La mayor parte del rendimiento de las arquitecturas profundas mencionadas se podía explicar por el desbalance en el grado de nodo de los datos. Tras permutar los pares positivos, manteniendo el grado de cada proteína, pero perdiendo el significado biológico de la interacción, el rendimiento de los métodos no se veía afectado de manera significativa. El rendimiento se mantenía especialmente alto utilizando los datasets HUANG (Huang *et al.*, 2015) y PAN (Pan *et al.*, 2010). Esto es debido a que en estos datasets, los grados de nodo en los pares positivos y pares negativos seguían distribuciones distintas: mientras que el grado positivo seguía una ley potencial, el grado negativo presentaba una distribución uniforme. Además, la mayoría de las proteínas tenían anotaciones mayoritariamente en pares positivos o negativos.

Finalmente, el estudio comprobó que excluir las proteínas de entrenamiento del conjunto de validación y minimizar la similitud de secuencia entre los conjuntos de entrenamiento y validación provocaba que las predicciones de los métodos basados en aprendizaje profundo se volviesen aleatorias.

El estudio de Bernett *et al.* (2024) demuestra que la predicción de interacciones con aprendizaje profundo requiere un cuidadoso diseño de los conjuntos de entrenamiento y validación, a fin de evitar falsas expectativas debidas al data leakage. Para impulsar la correcta validación de nuevos modelos, desarrollaron un dataset *gold-standard* sin filtraciones de grado de nodo y mínimas fugas por similitud de secuencia. El diseño consistió en particionar el proteoma humano en tres conjuntos disjuntos P_0, P_1 y P_2, con mínima similitud de secuencia. A continuación, agruparon el conjunto completo de PPI humanas extraídas de la base de datos HIPPIE (Alanis-Lobato *et al.*, 2017) en tres subconjuntos INTRA_0 (validación), INTRA_1 (entrenamiento) e INTRA_2 (test), de forma que cada par de proteínas pertenecía a INTRA_i si y solo si ambas pertenecían a P_i. De esta forma, los conjuntos de entrenamiento, validación y test no comparten proteínas, y el data leakage debido a grado de nodo no es posible. Los pares negativos se muestrearon aleatoriamente de P_0, P_1 y P_2, garantizando que el grado negativo siguiese una distribución similar al grado positivo.

En conclusión, la predicción de interacciones basada en secuencia es todavía un campo abierto que requiere más investigación. En este estudio, busca establecer un método basado en aprendizaje profundo capaz de realizar predicciones de interacción proteína-proteína fundamentadas en patrones con significado biológico, cuidando el uso de datasets sin data leakage.

1.3.1. ProtBERT-BiGRU-Attention para la predicción de PPI

Desde la publicación de Bernett *et al.* (2024), ha surgido un nuevo modelo de predicción de PPI basado en el pLM ProtBERT. La arquitectura de Gao *et al.* (2024) utiliza ProtBERT ajustado por fine-tuning como extractor de características, seguido de una capa BiGRU para refinarlas y capturar dependencias de largo rango, y un mecanismo de atención para priorizar los aminoácidos más relevantes para la interacción. Finalmente, se realiza una suma ponderada de los embeddings de cada aminoácido de acuerdo a los pesos de atención calculados y una clasificación binaria con una capa densamente conectada con activación sigmoidal. La figura 1.2 muestra el esquema global de la arquitectura.

Los autores tuvieron en cuenta los problemas de data leakage reportados por Bernett

et al. (2024) y entrenaron este modelo en su dataset *gold-standard*. Adicionalmente, entrenaron el modelo en los datasets de *Saccharomyces cerevisiae* GUO (Guo *et al.*, 2008) y humano PAN (Pan *et al.*, 2010), que presentan problemas de data leakage. Como comparación utilizaron varios modelos del estado del arte en predicción de interacciones, entre ellos D-SCRIPT, Topsy-Turvy y PIPR. Los autores reportaron unos resultados excelentes para ProtBERT-BiGRU-Attention, con una accuracy de 0.919 en el dataset de Bernett y algo mejor en GUO y PAN (0.950 y 0.928), lo cual atribuyen al data leakage de estos datasets. En contraste, el resto de modelos obtuvieron una accuracy menor y similar en los tres datasets, de alrededor de 0.7-0.8.

Mediante un ensayo de ablación, los autores defendieron la efectividad de las capas BiGRU y de atención. La eliminación del mecanismo de atención provocó una bajada

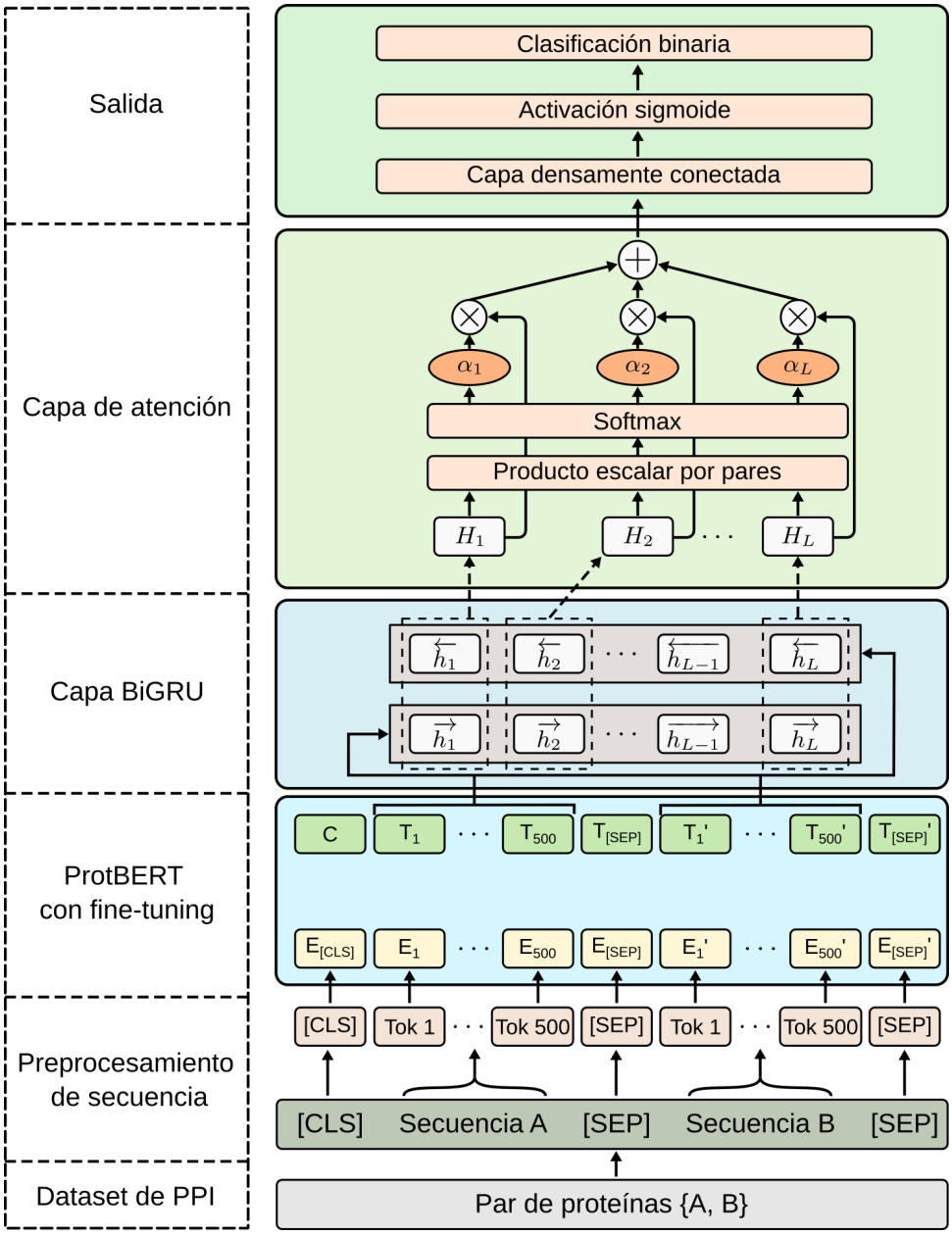

Figura 1.2: Arquitectura ProtBERT-BiGRU-Attention. Adaptado de Gao *et al.* (2024).

de 0.087 en la accuracy, mientras que la sustitución de la capa BiGRU por una GRU o BiLSTM provocó bajadas de 0.096 y 0.052 respectivamente. Los autores concluyeron que las capas BiGRU y atención resultaban eficaces para capturar información contextual de larga distancia en la secuencia y ponderar los aminoácidos de acuerdo a su contribución en la interacción. Además, comprobaron que ajustar ProtBERT con fine-tuning resultaba efectivo, y que este pLM presentaba un mejor rendimiento como tronco de la arquitectura frente a otros, como ProtT5 (Elnaggar *et al.*, 2022) o ESM2 (Lin *et al.*, 2023).

Pese a estos prometedores resultados, el método debe ser comprobado de forma rigurosa por expertos independientes para demostrar su efectividad. Su reproducibilidad resulta cuestionable, dado que el código de la implementación no se encuentra disponible. Este es un problema que afecta desgraciadamente a la investigación en aprendizaje automático, y dificulta la discusión científica (Kapoor & Narayanan, 2022). Además, el artículo presenta inconsistencias en sus resultados con los de Bernett *et al.* (2024). Extrañamente, no se observa el efecto del data leakage en los modelos utilizados como comparación. Todos ellos presentan una accuracy similar en los tres datasets, y los modelos D-SCRIPT, Topsy-Turvy y PIPR tuvieron resultados mejores que el azar en el dataset *gold-standard* de Bernett.

1.3.2. Modelos AlphaFold para la predicción de PPI

Aunque el problema del plegamiento en proteínas monoméricas fue resuelto por AlphaFold2, la predicción de la estructura de complejos multiméricos e interacciones proteína-proteína seguía siendo un gran desafío. Pese a estar fuera de su ámbito de entrenamiento, AlphaFold2 era capaz de predecir con buena precisión algunas interacciones multiméricas si se realizaban adaptaciones a la secuencia de entrada, mediante la adición de un linker flexible (Ko & Lee, 2021). Esta sorprendente capacidad de generalización motivó el desarrollo de AlphaFold-Multimer (Evans *et al.*, 2022), una versión específicamente diseñada para la predicción de complejos multiméricos, que superó a AlphaFold2 con entradas adaptadas y a los métodos basados en docking. Sin embargo, este modelo seguía presentando limitaciones, siendo por ejemplo incapaz de modelar la unión de anticuerpos a sus epítopos.

Recientemente, Google DeepMind ha desarrollado AlphaFold3 (Abramson *et al.*, 2024), que permite modelar las interacciones entre proteínas en complejos multiméricos con alta fidelidad. Este modelo presenta sustanciales mejoras de predicción frente a AlphaFold-Multimer, situándose a la vanguardia en la predicción de la estructura de interacciones proteína-proteína. Asimismo, es un método generalista, capaz de predecir la estructura de complejos que involucran otras biomoléculas no proteicas, como ligandos, iones, ácidos nucléicos, y proteínas con residuos modificados, tarea en la que también supera a métodos especializados de docking.

AlphaFold3 sufre un cambio fundamental en arquitectura frente a AlphaFold2 y AlphaFold-Multimer. Mientras que estos últimos realizan una tarea de regresión secuencia-estructura, AlphaFold3 es un modelo generativo condicionado en información de secuencia. Su arquitectura está formada por un modelo de difusión donde la mayor parte del procesamiento ocurre en el condicionamiento (figura 1.3). De forma similar a sus antecesores, recibe como entrada un MSA para cada proteína o cadena de ARN y opcionalmente, estructuras conocidas de proteínas homólogas (*templates*). Estos son embebidos en una representación individual (*single representation*) de ta-

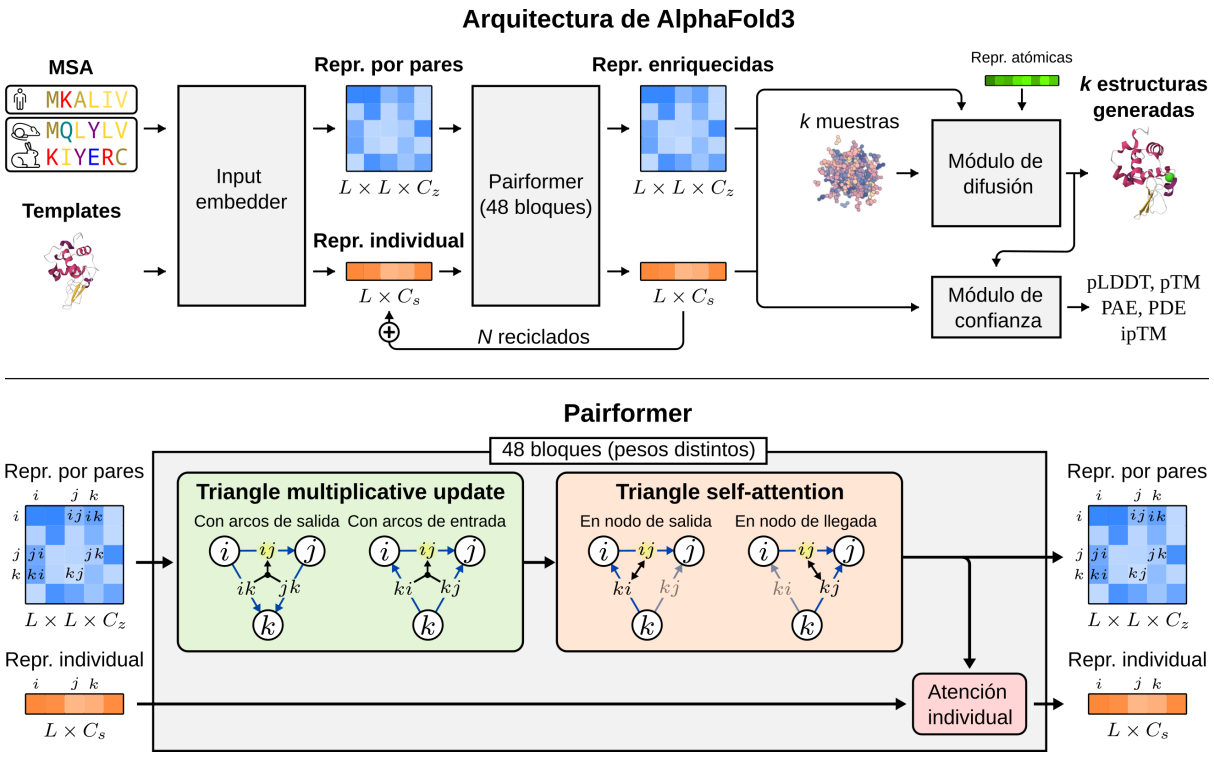

Figura 1.3: Arquitectura de AlphaFold3.

maño $L \times C_s$ y una representación por pares (*pair representation*) de tamaño $L \times L \times C_z$, donde L es el número de tokens de entrada, y C_s y C_z son los canales de cada representación. En AlphaFold3, cada token corresponde a un residuo de proteína, un nucleótido de ARN o un átomo pesado (no hidrógeno) de ligandos e iones. El MSA de entrada provee información coevolutiva entre residuos de la cadena polipeptídica. Esta información resulta clave para predecir la estructura de la proteína a partir de su secuencia. El fundamento es que si dos residuos de una proteína tienden a mutar juntos a lo largo de la evolución (coevolucionan), probablemente se encuentren físicamente cerca en la estructura tridimensional. Adicionalmente, la aportación de estructuras conocidas de proteínas similares mediante *templates* puede enriquecer las representaciones con información estructural adicional, aunque se ha demostrado que el MSA resulta suficiente para conseguir buenas reconstrucciones de la estructura (Jumper *et al.*, 2021).

A continuación, la representación individual y por pares se procesan en el módulo pairformer, que consiste en 48 bloques y las enriquece con relaciones espaciales (figura 1.3). Para ello, la representación por pares se interpreta como las características de los arcos de un grafo completamente conexo, donde cada token es un nodo. Cada bloque del pairformer modela las interacciones transitivas entre tokens, utilizando las características de los arcos ik y kj para actualizar aquellas del arco ij (*triangle multiplicative update*). La intuición tras este procesamiento proviene de AlphaFold2, y su objetivo es forzar el cumplimiento de la desigualdad triangular. De esta forma, las características pueden representar una distancia físicamente plausible en el espacio métrico. A continuación, se aplica un mecanismo de atención triangular (*triangle self-attention*), donde cada arco atiende a los demás arcos salientes o entrantes del nodo. Finalmente, la información de la representación por pares se añade a la representa-

ción individual durante una operación de atención. El embebido de MSA y templates, junto al procesamiento por el pairformer, se realiza repetidas veces en una técnica llamada recycling, lo que permite acumular y perfeccionar las representaciones.

Finalmente, las representaciones de tokens enriquecidas, junto con información atómica de entrada (enlaces covalentes o ligandos) se utilizan para condicionar la salida del módulo de difusión. Este ha sido entrenado para reconstruir las coordenadas reales de la estructura, a partir de versiones con ruido añadido. Los pasos de entrenamiento con poco ruido permiten perfeccionar el modelado de la estructura local (estereoquímica de los residuos), mientras que con mucho ruido se aprende a modelar la estructura global del complejo. Durante la inferencia, se utiliza ruido aleatorio como entrada, y se aplica el módulo recursivamente hasta obtener la estructura final. Una gran ventaja que provee este enfoque generativo es que la salida del modelo es una distribución de respuestas, en lugar de la única del enfoque regresivo de AlphaFold2 o AlphaFold-Multimer. Aplicando el módulo de difusión a varias entradas distintas de ruido aleatorio, se pueden obtener diferentes muestras de la estructura final. Esto permite disminuir la complejidad de AlphaFold3, ya que hace innecesario establecer restricciones de plausibilidad química en la estructura resultante y permite manejar la alta complejidad estructural de los ligandos, que pueden tener cualquier composición química (Abramson *et al.*, 2024).

AlphaFold3, igual que sus antecesores, provee métricas de confianza sobre la calidad de la estructura generada. Para ello, presenta un módulo de estimación de la confianza, que se entrena para aproximar distintas métricas de similitud de la estructura generada frente al *ground truth*, utilizando las representaciones de tokens enriquecidas y la estructura generada por el módulo de difusión en inferencia como entradas (figura 1.3). Las métricas de confianza globales disponibles son pLDDT (*predicted LDDT*), pTM (*predicted TM-score*), PAE (*predicted aligned error*) y PDE (*distance error matrix*). Asimismo, dado el propósito de modelar interacciones proteína-proteína, se provee una métrica que evalúa específicamente la interfaz de interacción: ipTM (*predicted interface TM-score*). pLDDT y pTM aproximan LDDT (*local-distance difference test*) y TM-score (*Template Modeling score*), dos medidas de similitud estructural entre proteínas. LDDT mide la similitud de dos estructuras en función del número de distancias locales entre átomos conservadas (Mariani *et al.*, 2013). TM-score, por su parte, mide la distancia entre los átomos Cα de residuos correspondientes en distintas superposiciones de las estructuras (Zhang & Skolnick, 2004). Ambos varían entre 0 y 1, con valores mayores indicando una mayor similitud. El ipTM es una variante del pTM, que únicamente tiene en cuenta en el cálculo pares de residuos en distintas cadenas. De esta forma, el ipTM solo evalúa la posición relativa entre distintas cadenas, excluyendo la calidad de reconstrucción de cada cadena por separado. Las matrices PAE y PDE proveen medidas de confianza entre pares de tokens. El elemento PAE_{ij} predice el error en la posición relativa del token j respecto al token i entre la estructura predicha y la real. El elemento PDE_{ij} mide el error en la distancia absoluta entre los tokens i y j.

En su artículo, los autores muestran que las métricas de confianza están bien calibradas con la calidad del modelo. El pLDDT de cada cadena proteica, de ARN o ligando muestra una clara correlación con el LDDT entre la predicción y la estructura real. A su vez, el ipTM y PAE entre cadenas, son buenos predictores de la calidad de reconstrucción de la interacción proteína-proteína, estando correlacionados con la métrica DockQ. DockQ es una medida comúnmente utilizada en la evaluación de mo-

delos de docking proteína-proteína, que proporciona una valoración continua entre 0 y 1, donde los valores más altos indican una reconstrucción del complejo proteico más precisa (Basu & Wallner, 2016). Un valor de ipTM superior a 0.8 se considera una predicción de alta calidad, asociada a DockQ mayores de 0.23, que indican una correcta posición relativa entre las cadenas. Valores de ipTM inferiores a 0.6 se consideran predicciones incorrectas de la interacción y la posición relativa de las cadenas no será correcta. Entre 0.6 y 0.8 existe una gran variabilidad en el DockQ, por lo que no se puede asegurar la fiabilidad de la predicción (Abramson *et al.*, 2024).

En resumen, AlphaFold3 supone un nuevo estado del arte en la predicción de complejos multiméricos de proteínas y provee métricas de confianza fiables.

1.4. Objetivos

El objetivo de este trabajo fue desarrollar un método de predicción de PPI entre proteínas basado únicamente en su secuencia, garantizando que resuelve la tarea correctamente y su rendimiento no se debe al data leakage reportado por Bernett *et al.* (2024). Para ello, se emplearon dos aproximaciones: una basada en el pLM ProtBERT y otra basada en AlphaFold3. La primera buscó replicar los prometedores resultados obtenidos por ProtBERT-BiGRU-Attention, así como evaluar el rendimiento del uso aislado de ProtBERT con fine-tuning para esta tarea. El empleo de ProtBERT era la opción más adecuada en términos de eficiencia computacional. Gracias a su menor número de parámetros y arquitectura de menor complejidad, presenta un tiempo de inferencia reducido frente a AlphaFold3. Además, no requiere el preprocesamiento de un MSA a partir de las secuencias, lo cual supone un gran cuello de botella en la inferencia con AlphaFold3. Debido a esto, un modelo basado en ProtBERT sería ideal para la predicción masiva de PPI, necesaria para completar con nuevos arcos grandes redes de interacciones entre proteínas.

Por otro lado, AlphaFold3 resulta más prometedor para la predicción de PPI, ya que se ha demostrado capaz de generar la estructura de complejos multiméricos a partir de su secuencia. Resulta razonable pensar que obtener una representación interna de la estructura a partir de la secuencia sea un paso crucial en la predicción de interacciones proteína-proteína con aprendizaje profundo. Como se ha comentado anteriormente, la estructura de una proteína es un factor determinante en su capacidad para establecer PPI, y por tanto, su buen entendimiento de la estructura provee a AlphaFold3 una gran ventaja frente a ProtBERT. Resulta importante destacar que, aunque en su artículo original se demostró que AlphaFold3 puede modelar correctamente la estructura de complejos multiméricos, no se ha comprobado que tenga la capacidad de distinguir entre proteínas que interactúan y las que no. En este trabajo, se busca dotar a AlphaFold3 de esta nueva capacidad para predecir interacciones proteína-proteína mediante una clasificación binaria. Para ello, se prueban dos opciones: buscar una asociación entre las métricas de calidad de interfaz y la presencia de interacción, y utilizar las representaciones por tokens enriquecidas como entrada a un modelo clasificador simple.

Capítulo 2

Métodos

2.1. ProtBERT-BiGRU-Attention

Se replicó la arquitectura de ProtBERT-BiGRU-Attention con la descripción que Gao *et al.* (2024) proveen en su artículo. Sin embargo, esta no era del todo completa, lo que ha llevado a realizar asunciones en algunos casos. Estas se indican cuando procede en el texto. La figura 2.1 resume el método utilizado en el entrenamiento de esta arquitectura.

2.1.1. Dataset

De la misma forma que en el artículo original, se utilizó el dataset *gold-standard* de Bernett para eliminar la posibilidad de un rendimiento sobreestimado debido a data leakage por el grado de nodo o similitud de secuencia. La versión 4 del dataset de Bernett fue descargada del enlace permanente provisto en su artículo (https://doi.org/10.6084/m9.figshare.21591618.v4). Este presenta tres particiones: INTRA_1 (entrenamiento, 163 192 pares de secuencias), INTRA_0 (validación, 59 260 pares de secuencias) e INTRA_2 (test, 52 048 pares de secuencias). Cada partición se encuentra balanceada, con el mismo número de pares positivos (PPI reales), que negativos (PPI falsas). No hay valores ausentes.

De forma similar a Gao *et al.* (2024), se realizó una división aleatoria de las particiones de entrenamiento, validación y test en un 90 % dedicado al fine-tuning del modelo ProtBERT y un 10 % dedicado al entrenamiento de las capas BiGRU y de atención.

2.1.2. Preprocesamiento de las secuencias

Se fijó la longitud de las secuencias a 500 residuos. Para ello, a las secuencias de menor longitud se les añadieron tokens de padding al final. En el caso de las secuencias de mayor longitud, se tomaron y concatenaron los primeros 250 aminoácidos del extremo N-terminal (inicio) y los últimos 250 aminoácidos del extremo C-terminal (final) para obtener secuencias de 500 aminoácidos (figura 2.1). Los códigos ambiguos de aminoácidos (U, Z) y aminoácidos poco comunes (O y B), fueron sustituidos por el comodín X, de acuerdo al método en que se entrenó ProtBERT (Elnaggar *et al.*, 2022). En ProtBERT, cada aminoácido es tratado como un token para el procesamiento por la arquitectura transformer. Dado que es un modelo BERT reentrenado, sus entradas

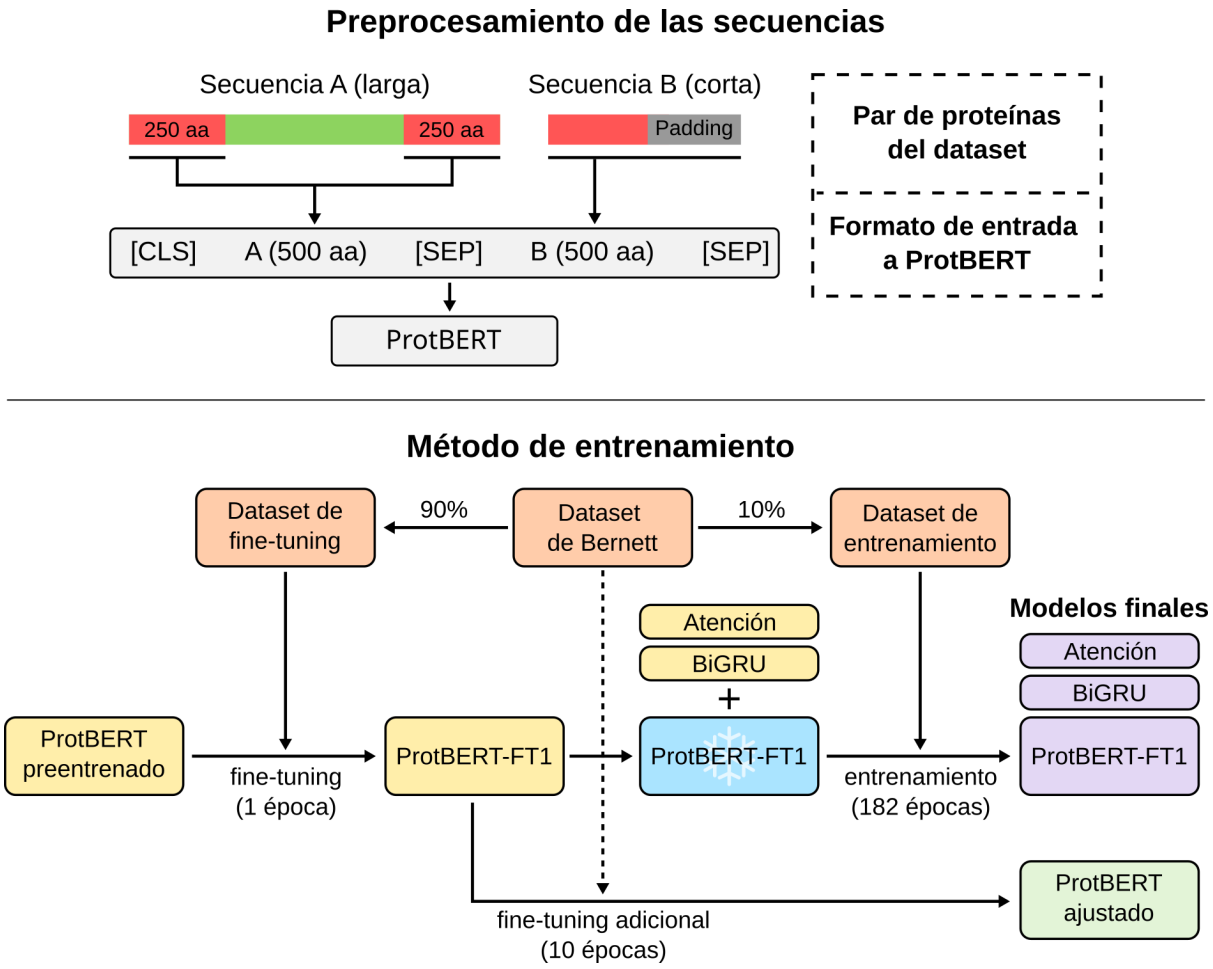

Figura 2.1: Metodología para el entrenamiento de la arquitectura ProtBERT-BiGRU-Attention y ProtBERT aislado con fine-tuning. En amarillo, los módulos entrenados en cada paso. En azul, los módulos con pesos congelados.

siguen el mismo formato. Al igual que en BERT, se deben añadir dos tokens especiales, el token clasificador (CLS) y el token separador (SEP). El token clasificador se añade siempre al comienzo de la secuencia de entrada, y su objetivo original es usar su estado oculto final como representación de la secuencia completa para tareas de clasificación (Devlin *et al.*, 2019). Los pares de secuencias se empaquetan en una única entrada, ambas delimitadas por el token separador. Para la clasificación de pares de proteínas potencialmente interactoras tokenizamos las secuencias siguiendo este formato:

[CLS] Secuencia de proteína 1 [SEP] Secuencia de proteína 2 [SEP]

La tokenización se realizó empleando la clase BertTokenizerFast de la librería transformers de HuggingFace, con la configuración del modelo ProtBERT preentrenado.

2.1.3. Fine-tuning de ProtBERT

El modelo ProtBERT preentrenado (https://huggingface.co/Rostlab/prot_bert) se ajustó mediante fine-tuning en el 90 % reservado del dataset de entrenamiento utilizando las clases BertForSequenceClassification y Trainer de la librería transformers de Hug-

gingFace. Dado que Gao *et al.* (2024) no especificaron el tipo de pooling utilizado para agrupar los estados ocultos finales de la pila de encoders para la entrada de la cabeza de clasificación, se decidió utilizar el método por defecto de la clase BertForSequenceClassification: el estado oculto final del token clasificador se hace pasar por una cabeza de clasificación binaria formada por una FFNN (*feed-forward neural network*) de dos capas, con dos salidas y activación softmax. Siguiendo el artículo original, se entrenaron únicamente la última capa de la pila de encoders y el clasificador, manteniendo el resto de pesos congelados. Se utilizó la entropía cruzada como función de coste.

El artículo de Gao *et al.* (2024), tampoco provee información sobre los hiperparámetros utilizados para el fine-tuning, por lo que se optó por valores estándar para modelos BERT (Devlin *et al.*, 2019). El entrenamiento se realizó durante tres épocas, con un tamaño de batch de 64, utilizando el optimizador AdamW con una tasa de aprendizaje de $2 \cdot 10^{-5}$ y un weight decay de 0.01. El resto de hiperparámetros se mantuvieron en los valores por defecto definidos por la clase Trainer de HuggingFace. El entrenamiento se aceleró mediante CUDA con una GPU NVIDIA A100 (40GB). Se evaluó la calidad del modelo en cada época según la accuracy en validación y se tomó para los siguientes pasos aquel que la maximizaba. El mejor modelo se alcanzó con la primera época (ProtBERT-FT1). La partición de test no se utilizó en este paso, ya que no se realizó ninguna optimización de hiperparámetros.

2.1.4. Implementación BiGRU

Tras el fine-tuning de ProtBERT, la cabeza de clasificación inicial se descartó y se acoplaron una capa BiGRU y de atención, seguidas de una nueva cabeza de clasificación binaria. La implementación de estas nuevas capas se realizó en el framework Pytorch 2.7.0. Siguiendo el diseño original, la capa BiGRU tiene como entrada los estados ocultos finales correspondientes a los tokens de aminoácidos, excluyendo los tokens especiales CLS y SEP. De esta forma, la entrada de la capa BiGRU es $X \in \mathbb{R}^{B \times L \times d}$, donde B es el tamaño de batch; $L = 1000$, la longitud de la secuencia y $d = 1024$, la dimensión de los estados ocultos de ProtBERT.

La arquitectura GRU es un tipo de red neuronal recurrente (RNN). Las RNN son capaces de procesar información secuencial de longitud arbitraria y capturar dependencias a lo largo de la secuencia, gracias a que utilizan como entrada no solo el dato actual, sino también el estado oculto del instante de tiempo anterior. Aplicando la red de forma recurrente en cada elemento de la secuencia, el estado oculto transferido actúa como una memoria que informa del contexto previo al procesar cada nuevo elemento de la secuencia. La arquitectura GRU emplea un modelo basado en una puerta de actualización y una puerta de reinicio, cuyo objetivo es añadir y eliminar información selectivamente del estado oculto.

El proceso de cálculo para un instante de tiempo $t \in \{1, 2, \ldots, L\}$, con vector de entrada $x_t \in \mathbb{R}^d$ es el siguiente:

$$
\begin{aligned}
z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z) \\
r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r) \\
\tilde{h_t} &= tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \\
h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h_t}
\end{aligned}
$$

donde h_t y $h_{t-1} \in \mathbb{R}^e$ representan los estados ocultos de los instantes t y $t-1$, $z_t \in (0,1)^e$ y $r_t \in (0,1)^e$ representan las puertas de actualización y reinicio, y $W \in \mathbb{R}^{e \times d}$, $U \in \mathbb{R}^{e \times e}$ y $b \in \mathbb{R}^e$ son pesos aprendidos. σ y $\odot$ representan la función de activación sigmoide y el producto de Hadamard, respectivamente.

La arquitectura BiGRU (GRU bidireccional) combina dos capas GRU que procesan desde el principio y final de la secuencia, lo cual permite tener en cuenta el contexto previo y futuro en cada instante de tiempo. Las salidas de ambas capas, $\overrightarrow{h_t}$ y $\overleftarrow{h_t}$ se combinan en un único estado oculto final H_t de la siguiente forma:

$$H_t = f(W_{\overrightarrow{h}} \overrightarrow{h_t} + W_{\overleftarrow{h}} \overleftarrow{h_t} + b_H),$$

donde $W \in \mathbb{R}^{e \times e}$ y $b \in \mathbb{R}^e$ son pesos aprendidos.

Para la implementación se utilizó el módulo GRU de Pytorch, con la opción bidireccional. La concatenación de los estados ocultos $\overrightarrow{h_t}$ y $\overleftarrow{h_t}$ se hace pasar por una capa lineal con activación ReLU de la siguiente forma:

$$H_t = ReLU(W_H[\overrightarrow{h_t}; \overleftarrow{h_t}] + b_H), \quad W_H \in \mathbb{R}^{e \times 2e}$$

Este cálculo resulta equivalente a sumar las proyecciones de cada estado oculto por separado.

2.1.5. Implementación del mecanismo de atención

Gao *et al.* (2024) emplean un mecanismo de atención propio, que reduce el conjunto de estados ocultos H_t a una representación compacta V de la secuencia. Esto se consigue mediante una combinación lineal de los estados ocultos H_t, donde los coeficientes α_t se calculan mediante un mecanismo de *self-attention*.

$$V = \sum_{t=1}^{L} \alpha_t H_t$$

El cálculo de los coeficientes se realiza a partir de una proyección de los estados ocultos:

$$u_t = tanh(W_u H_t + b_u)$$

Los coeficientes reflejan la similitud global de cada u_t al resto de instantes temporales u_w, calculada mediante el producto escalar y normalizada:

$$\alpha_t = \frac{\sum_w \exp(u_t \cdot u_w)}{\sum_i \sum_w \exp(u_i \cdot u_w)}$$

Hasta donde he podido investigar, este no es un mecanismo estándar de atención, por lo que tuvo que ser implementado desde cero en Pytorch. La descripción proporcionada por Gao *et al.* (2024) es numéricamente inestable. Además, durante su elección de hiperparámetros, mencionan el uso de varias cabezas de atención, sin

mostrar cómo se refleja esta modificación en el cálculo. Por tanto, para llevar a cabo la implementación, su mecanismo de atención tuvo que ser reformulado.

En primer lugar, se calcula la proyección lineal u_t a partir de los estados ocultos H_t:

$$u_t = W_u H_t + b_u, \quad W_u \in \mathbb{R}^{(n \cdot D_h) \times e}, b_u \in \mathbb{R}^{n \cdot D_h}$$

donde n es el número de cabezas de atención y D_h es la dimensión de cada cabeza de atención. La proyección u_t se redimensiona y divide en tensores $u_t^{(c)} \in \mathbb{R}^{D_h \times e}$ para cada cabeza de atención c. Por cada cabeza de atención, se realiza el cálculo de similitud con el producto escalar, modificado para garantizar la estabilidad numérica en la exponenciación:

$$\begin{aligned} a_{ij}^{(c)} &= u_i^{(c)} \cdot u_j^{(c)}, \quad i, j \in \{1, \ldots, L\}, c \in \{1, \ldots, n\} \\ \tilde{a}_{ij}^{(c)} &= a_{ij}^{(c)} - \max_{i,j,c} a_{ij}^{(c)} \end{aligned}$$

Los coeficientes de atención se calculan de la siguiente forma:

$$\alpha_t = \frac{\sum_{j,c} \exp(\tilde{a}_{tj}^{(c)})}{\sum_{i,j,c} \exp(\tilde{a}_{ij}^{(c)})}$$

Finalmente, la representación $V \in \mathbb{R}^e$ de la secuencia se obtiene con la combinación lineal:

$$V = \sum_{t=1}^{L} \alpha_t H_t$$

2.1.6. Implementación del clasificador

La salida del modelo se obtiene aplicando una capa lineal densa con dos salidas y activación softmax a la representación de secuencia V, para obtener una clasificación binaria.

$$Y = \text{softmax}(W_o V + b_o), \quad W_o \in \mathbb{R}^{2 \times e}, b_o \in \mathbb{R}^2$$

Esta cabeza de clasificación difiere de la empleada en el artículo original, donde usan una capa densa con una sola salida y activación sigmoidal. Este cambio se realizó debido a que el entrenamiento con softmax y entropía cruzada como función de coste podría resultar más estable.

2.1.7. Entrenamiento de BiGRU-Attention

El entrenamiento del modelo se realizó utilizando la clase Trainer de la librería transformers de HuggingFace, con aceleración CUDA en una GPU NVIDIA A100 (40GB).

Se empleó el modelo ProtBERT de la fase de fine-tuning con mejor accuracy en validación (ProtBERT-FT1), acoplado a las capas de BiGRU, atención y clasificación. Únicamente estas últimas se entrenaron, y los pesos de ProtBERT-FT1 se mantuvieron congelados (figura 2.1). El dataset de entrenamiento y validación utilizados en esta fase correspondían al 10 % del dataset *gold-standard* de Bernett como se indicó anteriormente.

En su artículo, Gao *et al.* (2024) emplean optimización bayesiana para determinar los hiperparámetros de entrenamiento óptimos. En este trabajo no se replicó esta sección debido al alto coste computacional que implica entrenar el modelo múltiples veces. En su lugar, se decidió reutilizar los hiperparámetros del artículo original cuando estaban disponibles. La tabla 2.1 muestra los valores utilizados. Dado que no se especifica en el artículo, el dropout se aplicó a las activaciones de la capa de salida del módulo BiGRU. Adicionalmente, las matrices de pesos de las capas BiGRU, atención y clasificador se inicializaron con He uniforme y los sesgos a cero, con el fin de incrementar la estabilidad del entrenamiento. Se utilizó AdamW como optimizador en lugar de Adam.

Hiperparámetro	**Valor**	**Fuente**
Épocas	182	Gao *et al.* (2024)
Batch size	32	Propia. Limitado por VRAM.
Learning rate	0.0017	Gao *et al.* (2024)
Weight decay	0.0124	Gao *et al.* (2024)
Dimensión estado oculto de BiGRU (e)	41	Gao *et al.* (2024)
Dimensión de la cabeza de atención (D_h)	32	Propia
Cabezas de atención (n)	2	Gao *et al.* (2024)
Dropout	0.2403	Gao *et al.* (2024)

Tabla 2.1: Hiperparámetros utilizados en el entrenamiento de la arquitectura ProtBERT-BiGRU-Attention.

2.2. ProtBERT con fine-tuning

Además de la arquitectura ProtBERT-BiGRU-Attention, se buscó evaluar el rendimiento de ProtBERT de forma aislada, con un ajuste fino en el dataset de Bernett. Para ello, se partió de ProtBERT-FT1, y se llevaron a cabo 10 épocas adicionales de ajuste sobre la totalidad de los datos de entrenamiento (INTRA_1) del dataset de Bernett (figura 2.1). En este caso, se entrenó el modelo completo, sin congelar pesos.

El entrenamiento se realizó utilizando la clase Trainer de la librería transformers de HuggingFace, con un tamaño de batch de 16, utilizando el optimizador AdamW con una tasa de aprendizaje de $2 \cdot 10^{-5}$ y weight decay de 0.01. El resto de hiperparámetros se mantuvieron en los valores por defecto definidos por la clase Trainer. El entrenamiento se aceleró utilizando CUDA con una GPU NVIDIA A100.

2.3. AlphaFold3

2.3.1. Modelo

El código para la inferencia con AlphaFold3 se obtuvo del repositorio oficial de Google DeepMind (https://github.com/google-deepmind/alphafold3) y fue instalado como paquete junto a las dependencias especificadas en el archivo *dev-requirements.txt* en un entorno virtual de Python 3.11.12 administrado con MiniForge. Los pesos de AlphaFold3 fueron solicitados directamente a Google de acuerdo con sus términos de uso.

Durante la inferencia se declaran las variables de entorno recomendadas por la documentación:

```
XLA_FLAGS="--xla_gpu_enable_triton_gemm=false"
XLA_PYTHON_CLIENT_PREALLOCATE=true
XLA_CLIENT_MEM_FRACTION=0.95
```

Además, debido a la alta exigencia de VRAM con secuencias de entrada largas, se utiliza en algunos casos memoria unificada. Para ello, se sustituyen las variables de entorno anteriores por las siguientes, tal y como recomienda la documentación de AlphaFold3 para una GPU NVIDIA A100 (40GB):

```
XLA_FLAGS="--xla_gpu_enable_triton_gemm=false"
XLA_PYTHON_CLIENT_PREALLOCATE=true
TF_FORCE_UNIFIED_MEMORY=true
XLA_CLIENT_MEM_FRACTION=3.2
```

2.3.2. Preprocesamiento de las secuencias de entrada

Cada par de secuencias de proteínas cuya interacción se quería comprobar, se codificó en un archivo AlphaFold3 JSON individual, según el formato descrito en la documentación de AlphaFold3. Cada secuencia del par se codifica en una cadena de proteína única e independiente. De esta forma, se realiza una inferencia de AlphaFold3 por cada par de proteínas, donde se determina la estructura del complejo multimérico formado por dos subunidades de estas.

A continuación, es necesario completar estos archivos de entrada con un alineamiento múltiple de secuencia (MSA) y *templates* para cada cadena. El código de AlphaFold3 permite llevar a cabo fácilmente este proceso con la herramienta HMMER (Eddy, 2009). Para ello, es necesario descargar localmente las bases de datos indicadas en el repositorio de AlphaFold3, cuyo tamaño conjunto alcanza los 700 GB descomprimidas. Estas bases de datos son capturas de UniProt, Uniref90, BFD, MGnify, NCBI nucleotide, RNAcentral, Rfam y RCSB PDB que contienen la mayoría de secuencias proteicas, secuencias de RNA y estructuras conocidas, sobre las cuales se realizan los alineamientos para obtener el MSA y las *templates*.

HMMER utiliza métodos probabilísticos, basados en la comparación de perfiles HMM (modelos ocultos de Markov) para encontrar secuencias homólogas a la secuencia de consulta (Eddy, 2009). Este método presenta una alta sensibilidad en proteínas lejanamente emparentadas y baja tasa de falsos positivos, lo que lo hace popular para la creación de MSA. Sin embargo, después de tratar de utilizarlo, se comprobó

que resultaba demasiado lento para permitir el preprocesamiento de los cientos de pares de secuencias requeridos para determinar una asociación entre las salidas de AlphaFold3 y la presencia de PPI. Además, este elevado tiempo de preprocesamiento, del orden de horas por secuencia, impediría el uso práctico para la predicción de PPI a escala masiva en redes de interacciones. En su lugar, se decidió utilizar MMseqs2 (Steinegger & Söding, 2017), un método de búsqueda optimizado para una gran velocidad y que mantiene una buena sensibilidad. Además, presenta la ventaja de que los creadores proveen un servidor para el cómputo remoto de MSA y *templates* dedicado al proyecto ColabFold (Mirdita *et al.*, 2022), una iniciativa para la ejecución conveniente e intuitiva de AlphaFold2 y AlphaFold-multimer desde un cuaderno de Google Colab. El servidor utiliza versiones actualizadas de UniProt, RCSB PDB, BFD, MGnify y otras bases de datos metagenómicas, por lo que puede realizar alineamientos de proteínas similares a los que se obtendrían con las versiones recomendadas por AlphaFold3, sin el coste de la ejecución local. El alineamiento de secuencias de ARN no es posible, aunque no resulta un impedimento para este trabajo.

Para realizar el cómputo de MSA y *templates*, se desarrolló un script de Python que toma una carpeta de archivos AlphaFold3 JSON con cadenas de proteínas, consulta al servidor MMseqs2 (https://a3m.mmseqs.com) la lista de secuencias no redundantes y puebla cada cadena con su correspondiente MSA y *templates*. El código está basado en el proyecto ColabFold, con modificaciones para utilizar los mismos requisitos de selección de *templates* que AlphaFold3. Las *templates* se seleccionan en base al alineamiento con las secuencias del RCSB PDB. Aquellas secuencias que sean demasiado similares a la consulta (100 % de identidad con cobertura menor al 95 %), malos alineamientos (menos de 10 % de identidad) o alineamientos cortos (menos de 10 residuos) se excluyen. El resto se seleccionan en orden de calidad hasta completar el número de *templates* solicitado, sus estructuras en formato mmCIF se obtienen de RCSB PDB (https://www.rcsb.org), se extrae únicamente la cadena de interés y se añade al archivo AlphaFold3 JSON.

Los archivos AlphaFold3 JSON poblados con MSA y *templates* pueden ser utilizados directamente por el código de inferencia de AlphaFold3. Se decidió utilizar un número de cuatro templates por cada cadena de proteína, el valor por defecto de AlphaFold3.

2.3.3. Datasets

Se utilizaron dos conjuntos de datos. La figura 2.2 describe su diseño. El primero estaba formado por pares de secuencias del dataset de Bernett. Se tomó una muestra aleatoria sobre el dataset de entrenamiento (INTRA_1), filtrando solo aquellos pares en los que ambas proteínas eran monoméricas. Realizar este filtrado podía resultar adecuado, ya que algunas proteínas requieren formar un complejo con otras previamente a interactuar con su pareja. En el preprocesamiento de las entradas a AlphaFold3 se asume que cada par de proteínas interactúa de forma binaria y aislada, por lo que las interacciones que dependen de más proteínas podrían ser difícilmente detectables. Seleccionando únicamente proteínas monoméricas, se elimina esta posibilidad. Para ello, se consultaron en UniProt (https://www.uniprot.org/) aquellas proteínas humanas y de estructura monomérica revisadas manualmente, con la siguiente consulta a día de 27 de abril de 2025:

(organism_id:9606) AND (cc_subunit:monomer) AND (reviewed:true)

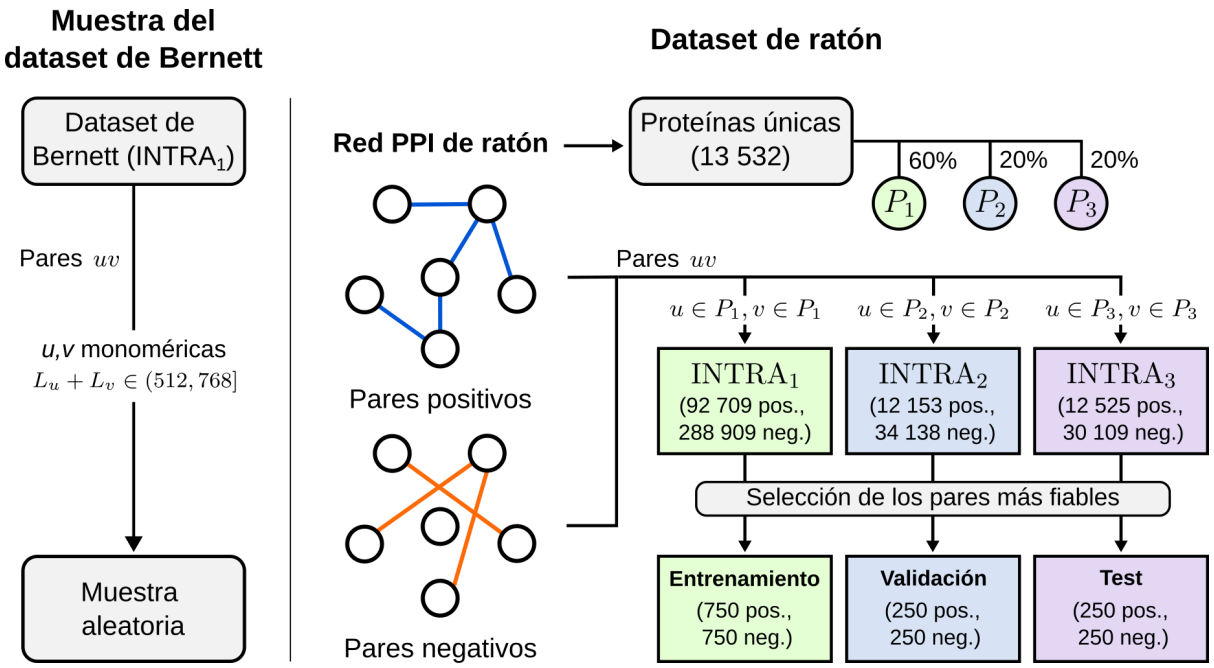

Figura 2.2: Diseño de los datasets utilizados para la validación de AlphaFold3 como método de predicción de PPI.

Se obtuvieron un total de 847 resultados. Se muestrearon aleatoriamente pares del dataset de entrenamiento de Bernett cuyas proteínas estuviesen incluidas en ellos. Adicionalmente, se seleccionaron únicamente aquellos pares en los que la suma de las longitudes de ambas proteínas, es decir, el número de tokens de entrada a AlphaFold3, se encontrase entre 512 y 768. La razón es que AlphaFold3 particiona sus longitudes de entrada en intervalos (*buckets*). Dentro de un mismo *bucket*, las secuencias de entrada se rellenan con tokens de *padding* hasta el límite superior, evitando una costosa recompilación del modelo. El *bucket* $(512, 768]$ contenía una cantidad representativa de pares de proteínas y tenía un tamaño suficientemente pequeño para una inferencia rápida.

El segundo conjunto de datos estaba formado por pares de secuencias de ratón provenientes de una red de PPI mantenida por el grupo de investigación. Al contrario que el dataset de Bernett, los pares de proteínas interactores preservaban una medida de la evidencia de interacción, por lo que se pudieron seleccionar solo aquellos cuya evidencia experimental de PPI directa era fuerte. La lista de pares negativos proviene del grafo complementario de la red de PPI. Se consideran de mayor fiabilidad y por tanto, se seleccionan preferentemente los pares de proteínas negativos con mayor distancia geodésica en la red de PPI. Se disponía de un total de 369 683 pares positivos y 1 045 945 pares negativos. Para limitar el tiempo de inferencia por cada par, se utilizaron únicamente aquellos en los que el número de tokens no excediese de 1536. De esta forma, se obtuvo un tiempo de inferencia máximo de aproximadamente 15 minutos por par en el hardware empleado. El dataset filtrado se componía de 278 300 pares positivos y 803 603 negativos.

Para prevenir el data leakage, se aplicó una estrategia de división del dataset de ratón similar a la usada por Bernett *et al.* (2024). Este paso resulta necesario para la predicción de PPI a partir de las representaciones internas de AlphaFold3, ya que

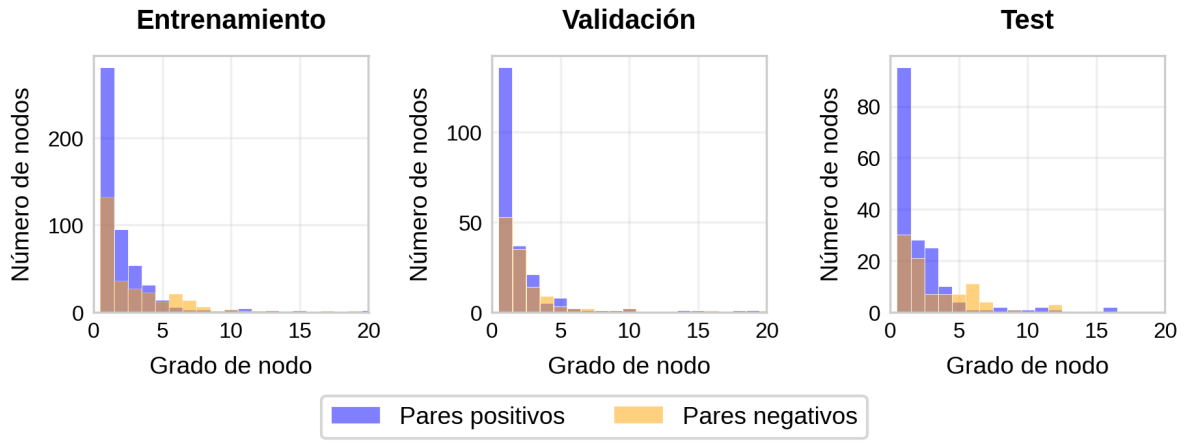

Figura 2.3: Distribución del grado de nodo positivo y negativo en los conjuntos del dataset de ratón.

se debe entrenar un modelo clasificador simple. Cualquier solape entre las proteínas que conforman los pares de entrenamiento, validación y test podría suponer que el modelo aprenda únicamente por grado de nodo. Aquellos pares que involucren una proteína que suela formar parte de pares positivos en entrenamiento, tenderán a ser predichos como positivos. El mismo fenómeno afecta a las predicciones negativas. Esto favorece que el modelo memorice las proteínas que actúan como nodos centrales en las interacciones positivas o negativas, en lugar de reconocer patrones relevantes para la interacción.

El conjunto de 13 532 proteínas distintas participantes en los pares positivos y negativos fue particionado aleatoriamente en tres bloques disjuntos, P_1, P_2 y P_3, en proporciones 60 %, 20 % y 20 % respectivamente. A continuación, los pares positivos y negativos se agruparon en los conjuntos INTRA_1 (92 709 positivos, 288 809 negativos), INTRA_2 (12 153 positivos, 34 138 negativos) e INTRA_3 (12 525 positivos, 30 109 negativos), de forma que el par $uv \in \text{INTRA}_i$ si y solo si $u \in P_i$ y $v \in P_i$. Finalmente, debido al alto coste computacional de inferencia con AlphaFold3, se seleccionaron subconjuntos de pares positivos y negativos con la mayor evidencia experimental de cada bloque INTRA para conformar los conjuntos de entrenamiento, validación y test. El conjunto de entrenamiento se consituyó con 750 pares positivos y 750 negativos de INTRA_1, el conjunto de validación con 250 pares positivos y 250 negativos de INTRA_2, y el conjunto de test con 250 pares positivos y 250 negativos de INTRA_3. Adicionalmente, se comprobó que el grado de nodo seguía una distribución similar para los pares positivos y negativos en los conjuntos de entrenamiento, validación y test (figura 2.3). Esto resulta relevante, ya que si las distribuciones fueran diferentes para pares positivos y negativos, el modelo podría encontrar patrones en el grado de nodo en los que basar su predicción.

2.3.4. Predicción de PPI con métricas de confianza

La inferencia con AlphaFold3 se realizó con los parámetros por defecto: 10 ciclos de reciclaje y 5 muestras de difusión por semilla aleatoria. Esto resulta en $5n$ estructuras de salida por cada entrada, donde n es el número de semillas. El código de inferencia ordena automáticamente las estructuras generadas según su calidad,

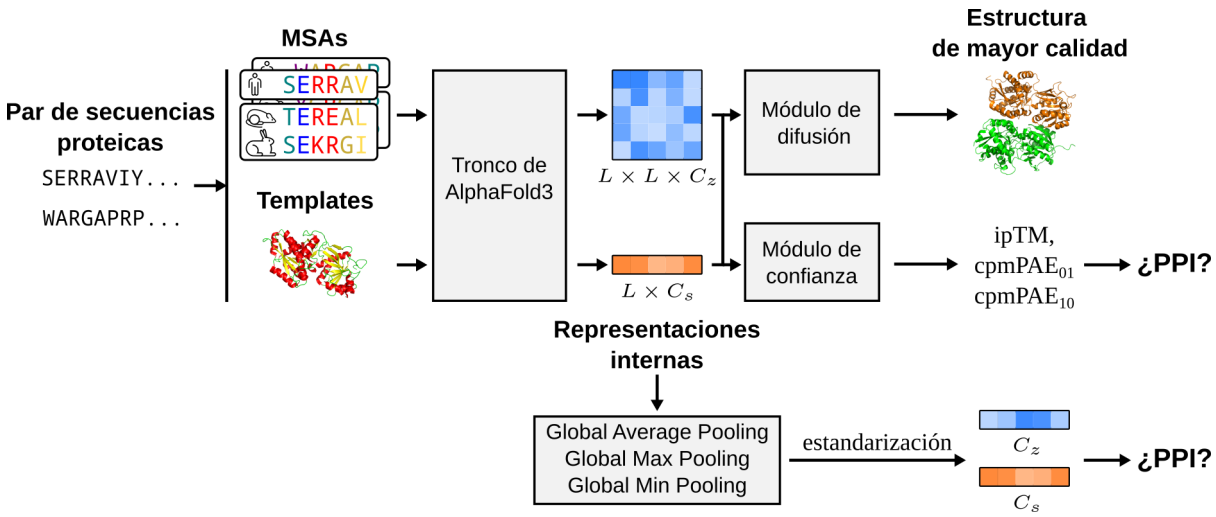

Figura 2.4: Aproximaciones para la predicción de PPI con AlphaFold3.

medida como un agregado de las métricas de confianza pTM, ipTM, la presencia de regiones desordenadas y colisiones estéricas. Para la predicción de PPI, se utilizaron las métricas de confianza del modelo de mayor calidad para cada par de proteínas. Se buscó una asociación entre las métricas de confianza ipTM y *chain pair PAE min*, y la presencia de PPI (figura 2.4). La métrica ipTM se utilizó ya que se ha demostrado que está correlacionada con DockQ, una medida de la calidad de modelado de la interfaz del complejo. Por otro lado, *chain pair PAE min* (cpmPAE) es el PAE mínimo entre residuos de las dos cadenas, y por tanto, es un predictor del error en su orientación relativa. Su definición es la siguiente:

$$\text{cpmPAE}_{XY} = \min_{i,j} \text{PAE}_{ij}, \quad i \in X, j \in Y$$

donde i y j son residuos pertenecientes a las cadenas X e Y, respectivamente.

La documentación de AlphaFold3 sugiere que esta métrica correlaciona con la presencia de PPI, y que en algunos casos es capaz de distinguir proteinas interactoras de las no interactoras. Sin embargo, en el artículo original solo se muestra una correlación del PAE entre cadenas con la calidad de reconstrucción de la interfaz de interacción (DockQ).

Para determinar su capacidad predictiva de PPI, se entrenaron modelos de regresión logística utilizando la librería scikit-learn 1.3.0, en ambas métricas de confianza de forma aislada y combinada. En el caso de *chain pair PAE min* se utilizaron como características de entrada únicamente los valores entre las dos cadenas diferentes del par: cpmPAE$_{01}$ y cpmPAE$_{10}$. Las regresiones logísticas se entrenaron con el optimizador L-BFGS, regularización L2 y coeficiente de regularización $C = 1$.

2.3.5. Predicción de PPI con representaciones internas

La inferencia se realizó con los parámetros por defecto de AlphaFold3: 10 ciclos de reciclaje y 5 muestras de difusión por semilla aleatoria. Adicionalmente, se solicitó almacenar las representaciones internas por tokens con la opción - -*save_embeddings*

del código de inferencia. Se utilizó una única semilla aleatoria, dando lugar a una única representación individual ($s \in \mathbb{R}^{L \times C_s}$) y por pares ($z \in \mathbb{R}^{L \times L \times C_z}$), donde L es el número de tokens de entrada. $C_s = 128$ y $C_z = 384$ son el número de canales de la representación individual y por pares, respectivamente.

A continuación, se obtuvieron representaciones de tamaño fijo a partir de las representaciones por pares e individual, mediante estrategias de agrupamiento global (*global pooling*). En concreto, se probaron *global average pooling*, *global max pooling* y *global min pooling*. Se realizó un estudio para determinar la combinación de representación de entrada (individual o por pares) y agrupamiento que mejor rendimiento de clasificación lograba.

Global average pooling toma un tensor $X \in \mathbb{R}^{d_1 \times \ldots \times d_n \times C}$ y da como resultado un vector $v \in \mathbb{R}^C$ tal que:

$$v_c = \sum_{i_1,\ldots,i_n} X_{i_1,\ldots,i_n,c}, \quad c \in \{1,\ldots,C\}$$

Global max pooling toma un tensor $X \in \mathbb{R}^{d_1 \times \ldots \times d_n \times C}$ y da como resultado un vector $v \in \mathbb{R}^C$ tal que:

$$v_c = \max_{i_1,\ldots,i_n} X_{i_1,\ldots,i_n,c}, \quad c \in \{1,\ldots,C\}$$

Global min pooling toma un tensor $X \in \mathbb{R}^{d_1 \times \ldots \times d_n \times C}$ y da como resultado un vector $v \in \mathbb{R}^C$ tal que:

$$v_c = \min_{i_1,\ldots,i_n} X_{i_1,\ldots,i_n,c}, \quad c \in \{1,\ldots,C\}$$

Seguidamente, cada canal c de las representaciones v_c fue reescalado a una media de cero y desviación típica de uno (estandarización):

$$v_c \leftarrow \frac{v_c - \mu_c}{\sigma_c}$$

donde μ_c y σ_c son la media y desviación típica del canal c en el conjunto de entrenamiento. Este proceso se esquematiza en la figura 2.4.

Finalmente, se utilizó una regresión logística (scikit-learn 1.3.0) como clasificador binario sobre las representaciones estandarizadas de tamaño fijo. El entrenamiento se realizó en el dataset de ratón (1500 ejemplos). El conjunto de validación (500 ejemplos) se empleó para realizar una optimización de hiperparámetros mediante búsqueda exhaustiva en el espacio mostrado en la tabla 2.2. El modelo con mayor accuracy en validación fue seleccionado, y evaluado en el conjunto de test (500 ejemplos).

Parámetro	**Valores**	**Notas**
Optimizador	L-BFGS, Newton-CG, SAG, SAGA	Regularización L2
	Liblinear, SAGA	Regularización L1
Regularización	L2, L1	Depende del optimizador
C	0.001, 0.01, 0.1, 1, 10	Inversa de la fuerza de regularización

Tabla 2.2: Espacio de búsqueda para la optimización de hiperparámetros de las regresiones logísticas aplicadas sobre las representaciones internas de AlphaFold3.

2.4. Acceso al código utilizado

El código empleado en la realización de este trabajo se encuentra alojado en el repositorio https://github.com/liaupm/PPI-prediction.

Capítulo 3

Resultados

3.1. ProtBERT-BiGRU-Attention

Se replicó el método de Gao *et al.* (2024) en el desarrollo de su arquitectura ProtBERT-BiGRU-Attention para la clasificación de PPI basada en secuencia (ver métodos). Para ello, en primer lugar, se realizó un fine-tuning de ProtBERT en el 90 % del dataset *gold-standard* de Bernett. Durante tres épocas, se entrenó únicamente el último encoder de la pila de ProtBERT, manteniendo el resto de pesos congelados. Al final de cada época, se evaluó el modelo según su accuracy en el conjunto de validación. El resultado del ajuste se muestra en la tabla 3.1. Como se puede observar, el rendimiento decreció ligeramente a lo largo de las iteraciones. El modelo con la mayor accuracy, de 0.547, se obtuvo con una sola época de ajuste. Por claridad, este modelo se denomina ProtBERT-FT1. Su rendimiento no difiere significativamente del azar (accuracy 0.5), por lo que el fine-tuning de ProtBERT ajustando únicamente la última capa resulta insuficiente para hacerlo útil como clasificador de PPI.

Época	Pérdida en validación	Accuracy en validación
1	0.689	0.547
2	0.691	0.545
3	0.694	0.541

Tabla 3.1: Evaluación del modelo ProtBERT en el conjunto de validación durante el fine-tuning.

A continuación, se tomó ProtBERT-FT1 como tronco de la arquitectura ProtBERT-BiGRU-Attention. Durante el entrenamiento, los pesos de ProtBERT se mantuvieron congelados, mientras que se actualizaban los de las capas BiGRU y de atención. Los resultados no fueron satisfactorios, por lo que se realizaron dos entrenamientos adicionales con algunas modificaciones. La figura 3.1 muestra las curvas de aprendizaje de los tres entrenamientos llevados a cabo.

El entrenamiento 1 se realizó en el 10 % restante del dataset de Bernett, con los hiperparámetros definidos en la tabla 2.1. La curva de aprendizaje mostró un claro patrón de overfitting al conjunto de entrenamiento. A lo largo de las 182 épocas, la función de pérdida en entrenamiento tendió a cero, mientras que la pérdida en validación disminuyó levemente durante las primeras diez épocas, momento a partir del cual aumentó sin cota. El rendimiento en validación se mantuvo cerca del azar, obtenién-

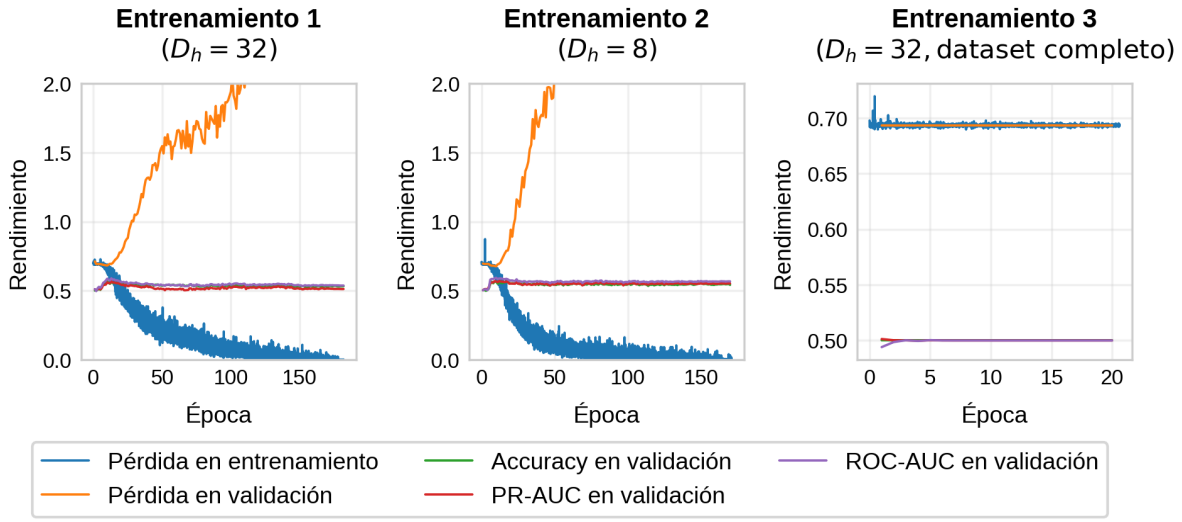

Figura 3.1: Curvas de aprendizaje de ProtBERT-BiGRU-Attention.

dose la mejor accuracy, de 0.566, en la décima época. Las métricas ROC-AUC (área debajo de la curva ROC) y PR-AUC (área bajo la curva precisión-sensibilidad) también mostraron un rendimiento similar al azar (ROC-AUC ≈ 0.5, PR-AUC ≈ 0.5) durante todo el entrenamiento. El overfitting tan temprano en el entrenamiento impedía un buen desempeño del clasificador. Por tanto, se diseñaron dos nuevos entrenamientos con el fin de disminuirlo.

El entrenamiento 2 se realizó en el mismo conjunto de datos, pero se disminuyó el tamaño de la cabeza de atención de $D_h = 32$, a $D_h = 8$. La motivación tras este cambio proviene de que este parámetro no había sido especificado en el artículo original, por lo que quizá Gao *et al.* (2024) utilizaron un número menor, disminuyendo el número de parámetros del modelo, y previniendo el overfitting en su caso. Sin embargo, esta modificación no resultó suficiente. La curva de aprendizaje mostró un patrón de overfitting similar al entrenamiento 1. El rendimiento se mantuvo cerca del azar, alcanzando una accuracy máxima de 0.567 en la novena época.

Finalmente, el entrenamiento 3 buscó aumentar el número de datos para evitar el overfitting. Se utilizaron los conjuntos de entrenamiento y validación completos del dataset de Bernett, suponiendo un incremento de diez veces en el tamaño del conjunto de entrenamiento respecto a los entrenamientos 1 y 2. Se utilizó el tamaño de cabeza de atención original $D_h = 32$. El resultado fue una ausencia total de aprendizaje, tanto en el conjunto de entrenamiento como en validación. La función de pérdida en entrenamiento y validación se mantuvo invariante a lo largo de veinte épocas. Asimismo, el rendimiento en validación fue indistinguible del azar, con una accuracy estable de 0.50. El entrenamiento se truncó a las veinte épocas, cuando se determinó que no era productivo.

Estos resultados sugieren que la arquitectura ProtBERT-BiGRU-Attention aquí empleada es incapaz de predecir PPI a partir de la secuencia. Su aprendizaje en los entrenamientos 1 y 2 se sustenta en la memorización del pequeño conjunto de entrenamiento. Cuando el conjunto de entrenamiento es suficientemente grande para impedir este sobreajuste, el modelo es incapaz de lograr ningún aprendizaje. Asimismo,

se observa que el modelo es equiparable a una clasificación aleatoria en validación. Sólo se logra un rendimiento en validación levemente mayor en los entrenamientos 1 y 2, que puede ser explicado por una leve similitud de algunos pares de secuencias a los de entrenamiento, ya que resulta imposible eliminar completamente el data leakage por similitud de secuencia.

3.2. ProtBERT con fine-tuning

A continuación, se buscó determinar la eficacia de ProtBERT de forma aislada para la clasificación de PPI en el dataset de Bernett. Para ello, se realizaron diez épocas adicionales de fine-tuning, partiendo de ProtBERT-FT1. En esta ocasión, se permitió entrenar la totalidad de los pesos del modelo, lo que ayudaría a aumentar su flexibilidad en el aprendizaje de patrones relevantes para la clasificación de PPI. El entrenamiento y validación se realizaron con el dataset completo de Bernett.

La curva de aprendizaje se puede observar en la figura 3.2. Los resultados fueron muy similares a los obtenidos durante el entrenamiento 3 del modelo ProtBERT-BiGRU-Attention (figura 3.1). Se observó una ausencia total de aprendizaje. La función de pérdida en entrenamiento y validación se mantuvo invariante, y el rendimiento en validación indistinguible del azar, con una accuracy constante de 0.50.

Estos resultados sugieren que ProtBERT es incapaz de extraer patrones relevantes para la clasificación de PPI a partir de secuencia, incluso permitiendo el ajuste de todos los pesos del modelo.

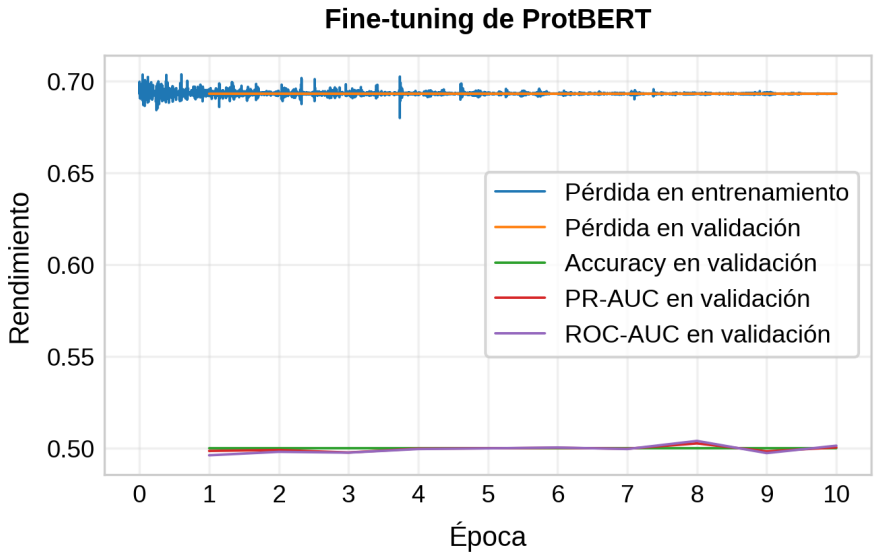

Figura 3.2: Curva de aprendizaje de ProtBERT con fine-tuning.

3.3. AlphaFold3

3.3.1. Predicción de PPI con métricas de confianza

Entre las aproximaciones para emplear AlphaFold3 como predictor de PPI, se decidió explorar el uso de las métricas de confianza ipTM y *chain pair PAE min*. Estas métricas están relacionadas con la calidad de modelado de la interfaz en la estructura del complejo generada por AlphaFold3, por lo que podrían ser predictoras de la presencia o ausencia de interacción. Esta asociación se buscó en dos conjuntos de datos: una muestra balanceada de 100 pares de proteínas monoméricas del dataset de Bernett y un dataset de ratón libre de data leakage diseñado a partir de una red de PPI mantenida por el grupo de investigación (1500 pares de entrenamiento, 500 de validación y 500 de test).

La estructura de los 100 pares de proteínas de la muestra del dataset de Bernett fue predicha con AlphaFold3. Por cada par, se utilizaron cinco semillas aleatorias y cinco muestras de difusión, dando lugar a 25 modelos de su estructura, entre los cuales se seleccionó el de mayor calidad. A continuación, se evaluó la capacidad predictiva del ipTM y *chain pair PAE min* de forma aislada y conjunta mediante el entrenamiento de regresiones logísticas con validación cruzada de 4 particiones. En el caso de *chain pair PAE min*, se utilizaron como características el mínimo PAE entre la primera y segunda cadena (cpmPAE_{01}) y entre la segunda y primera (cpmPAE_{10}), descartándose las medidas intracadena ya que eran constantes y no aportaban información sobre la interfaz.

Los resultados de la clasificación se muestran en la tabla 3.2. La accuracy media en validación cruzada correspondía a una clasificación aleatoria (accuracy ≈ 0.5), por lo que las métricas de confianza no mostraron capacidad predictiva de PPI. La figura 3.3 muestra además que ipTM y *chain pair PAE min* siguen distribuciones similares tanto en los pares de proteínas interactores como no interactores. Esto explica el bajo rendimiento de la clasificación y sugiere que estas métricas de confianza no se encuentran relacionadas con la presencia de PPI.

Características de entrada	**Accuracy media (4-fold CV)**
ipTM	0.450
cpmPAE_{01}, cpmPAE_{10}	0.490
ipTM, cpmPAE_{01}, cpmPAE_{10}	0.500

Tabla 3.2: Clasificación de PPI en base a las métricas de confianza ipTM y *chain pair PAE min* de AlphaFold3 en una muestra $n = 100$ balanceada del dataset de Bernett. Los resultados provienen de una validación cruzada de 4 particiones (4-fold CV) con regresión logística como clasificador.

Un método similar se siguió para el dataset de ratón. La estructura de los 2500 pares de proteínas fue predicha con AlphaFold3. Debido al mayor tamaño del dataset y el alto coste de inferencia, se utilizó una única semilla aleatoria y cinco muestras de difusión por par. Los valores de ipTM y *chain pair PAE min* del modelo de mejor calidad se utilizaron para entrenar modelos de regresión logística. Dado que no se pretendía realizar optimización de hiperparámetros, se fusionaron los pares de entrenamiento y validación en un único conjunto de entrenamiento (2000 pares). Los clasificadores se evaluaron en el conjunto de test (500 pares).

Los resultados fueron de nuevo insatisfactorios (tabla 3.3). Los clasificadores tuvie-

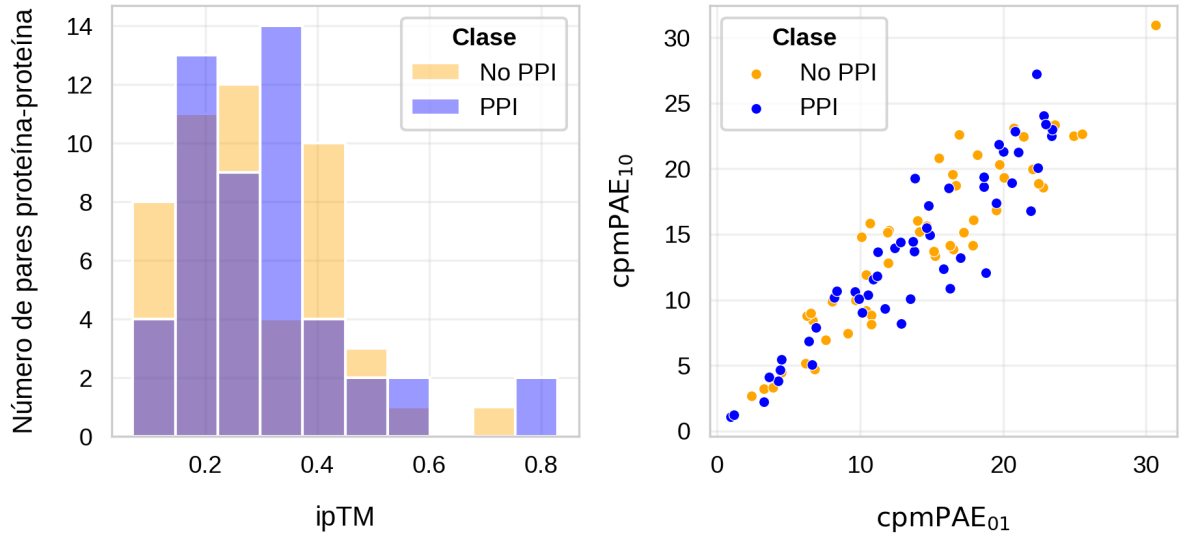

Figura 3.3: Distribución de las métricas de confianza ipTM y *chain pair PAE min* de AlphaFold3 para una muestra $n = 100$ balanceada del dataset de Bernett.

ron un rendimiento inferior al esperado por azar en el conjunto de test, lo cual es indicativo de que ajustaron patrones falsos en el entrenamiento. Las distribuciones del ipTM y *chain pair PAE min* muestran de nuevo una falta de asociación con la presencia de interacción (figura 3.4). Curiosamente, los pares de proteínas no interactores parecen presentar valores ligeramente mayores de ipTM, al contrario de lo que cabría esperar. Sin embargo, como muestran los resultados de clasificación, las distribuciones de pares interactores y no interactores siguen teniendo mucho solape para ser separables.

Características de entrada	**Accuracy**	**ROC AUC**	**PR AUC**
ipTM	0.412	0.269	0.458
cpmPAE_{01}, cpmPAE_{10}	0.346	0.304	0.442
ipTM, cpmPAE_{01}, cpmPAE_{10}	0.370	0.301	0.451

Tabla 3.3: Clasificación de PPI en base a las métricas de confianza ipTM y *chain pair PAE min* en el dataset de ratón. Los resultados provienen de regresiones logísticas entrenadas en la fusión del conjunto de entrenamiento y validación (2000 ejemplos) y evaluadas en el conjunto de test (500 ejemplos).

Los resultados sugieren que las métricas de confianza ipTM y *chain pair PAE min* de AlphaFold3 no son buenas características para la predicción de PPI. El bajo rendimiento de clasificación con estas métricas ocurre incluso en abundancia de ejemplos de entrenamiento y validación, como es el caso del dataset de ratón. Además, la visualización directa de las distribuciones sugiere que no existen diferencias en ipTM y *chain pair PAE min* entre pares de proteínas interactores y no interactores.

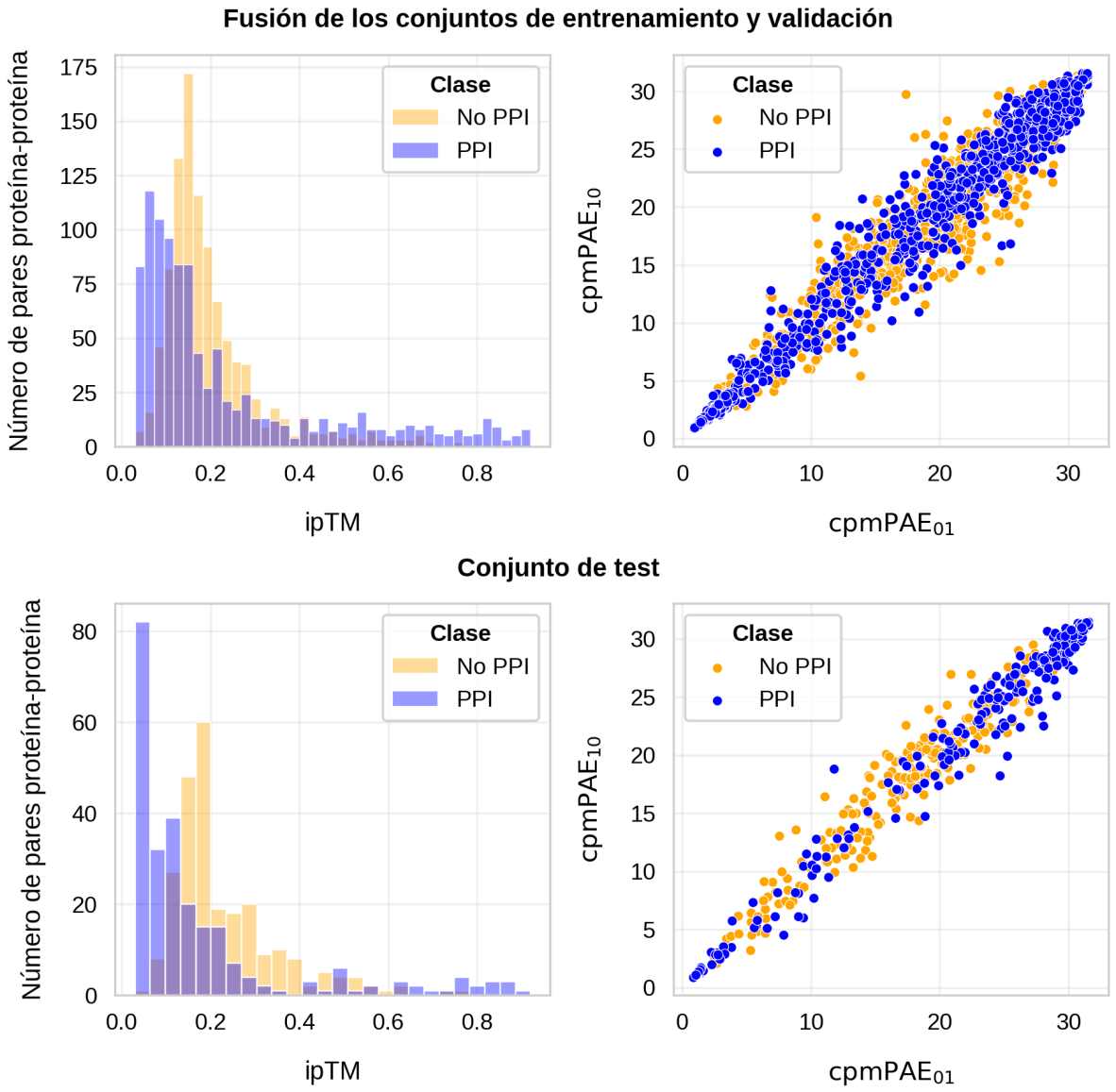

Figura 3.4: Distribución de las métricas ipTM y *chain pair PAE min* de AlphaFold3 en la fusión del conjunto de entrenamiento y validación (2000 pares proteína-proteína), y conjunto de test (500 pares) del dataset de ratón.

3.3.2. Predicción de PPI con representaciones internas

La segunda aproximación para emplear AlphaFold3 como predictor de PPI, pretendió utilizar las representaciones de tokens enriquecidas generadas a la salida del módulo pairformer. Dado que estas representaciones se utilizan directamente para la generación de las estructuras, deben contener información estructural del complejo proteico que podría ser indicadora de la presencia o ausencia de interacción. A fin de comprobar esta hipótesis, se decidió entrenar modelos de regresión logística en la tarea de clasificación de PPI a partir de las representaciones individual y por pares de AlphaFold3. Para ello, se utilizó el dataset de ratón (1500 pares de entrenamiento, 500 de validación y 500 de test). La dimensión de las representaciones se redujo a un tamaño fijo mediante tres tipos de agrupamiento: *global average pooling*, *global max*

pooling y *global min pooling*. A continuación, las representaciones se estandarizaron, y usaron como entrada a los modelos de regresión logística.

Para cada combinación de tipo de representación y agrupamiento, se realizó una optimización de hiperparámetros utilizando el conjunto de validación. La tabla 3.4 muestra el rendimiento obtenido en el conjunto de validación para cada combinación. Los resultados fueron significativamente mejores que el azar en todos los casos. El mejor modelo se obtuvo utilizando la representación individual con un *global average pooling* como entrada a la regresión logística. Este alcanzó una accuracy de 0.818, ROC-AUC de 0.907 y PR-AUC de 0.917.

Representación	Agrupamiento	Hiperparámetros	Accuracy	ROC AUC	PR AUC
Individual	Global Average	L-BFGS, reg. L2, $C = 0.01$	0.818	0.907	0.917
Individual	Global Max	L-BFGS, reg. L2, $C = 0.001$	0.778	0.875	0.865
Individual	Global Min	L-BFGS, reg. L2, $C = 0.001$	0.766	0.869	0.858
Por pares	Global Average	SAGA, reg. L2, $C = 10$	0.806	0.877	0.880
Por pares	Global Max	L-BFGS, reg. L2, $C = 0.001$	0.706	0.786	0.803
Por pares	Global Min	L-BFGS, reg. L2, $C = 0.001$	0.720	0.801	0.807

Tabla 3.4: Rendimiento en el conjunto de validación de distintas combinaciones de representación y agrupamiento para la clasificación de PPI con regresión logística a partir de las representaciones internas de AlphaFold3. Se muestran los resultados con hiperparámetros óptimos para cada combinación.

Para determinar su capacidad de generalización y descartar que este buen rendimiento se debiese a un sobreajuste de los hiperparámetros al conjunto de validación, el mejor modelo se evaluó en el conjunto de test. El modelo logró determinar la presencia de PPI con una accuracy alta de 0.840, ROC-AUC de 0.904 y PR-AUC de 0.930. La matriz de confusión y la curva precisión-sensibilidad se muestran en la figura 3.5. Esta última muestra el equilibrio entre precisión y sensibilidad según se varía el umbral de decisión $p = \alpha$ (donde p es la probabilidad de PPI predicha por la regresión

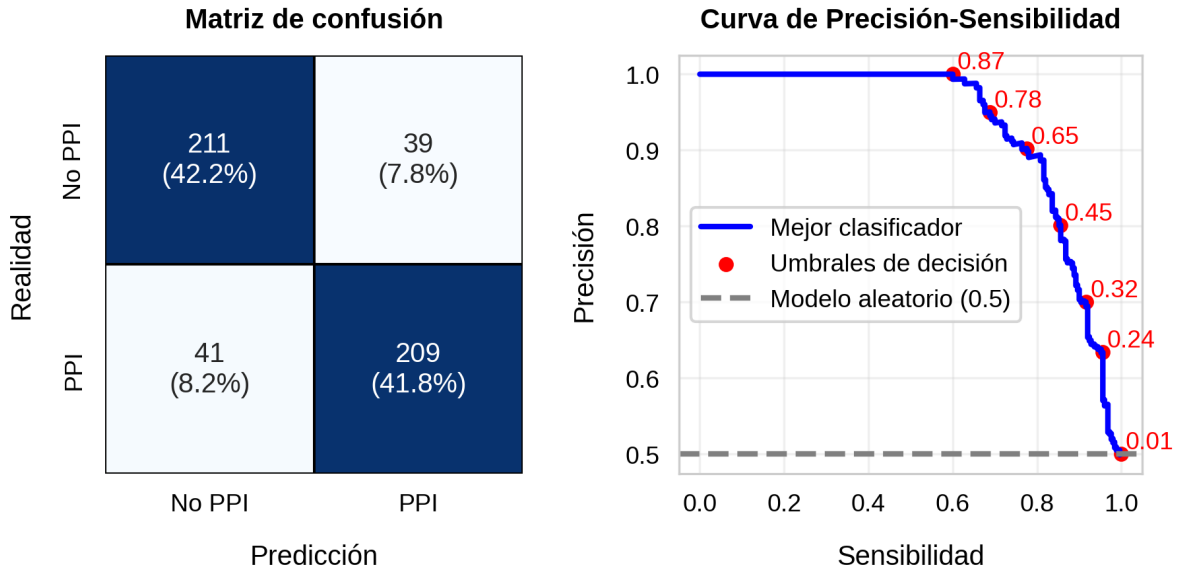

Figura 3.5: Matriz de confusión y curva precisión-sensibilidad del mejor clasificador de PPI basado en representaciones individuales de AlphaFold3.

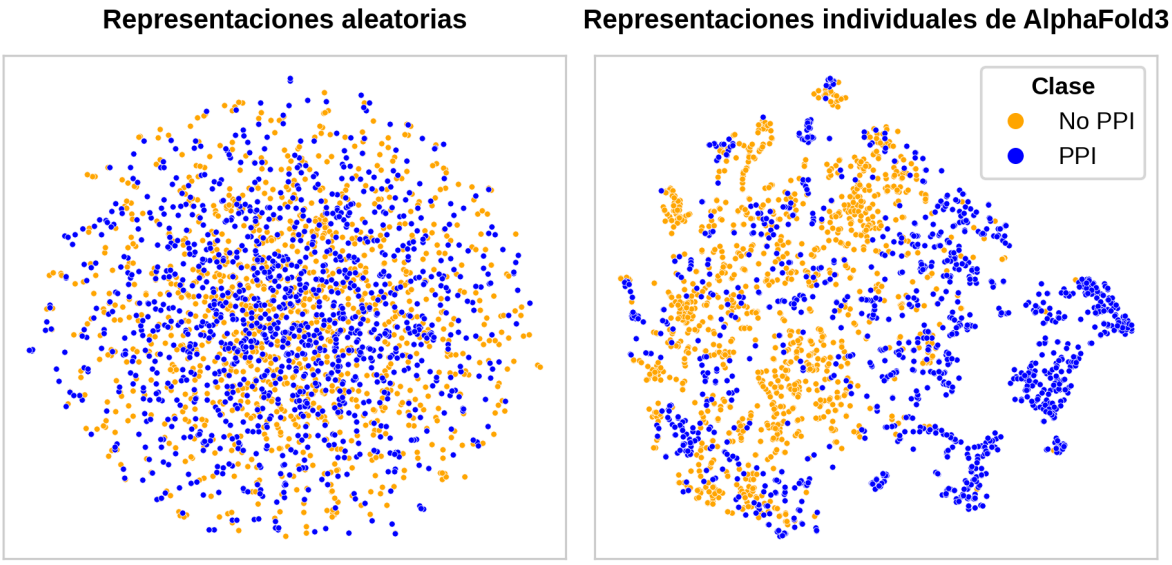

Figura 3.6: Visualización t-SNE de las representaciones individuales de AlphaFold3 para el dataset de ratón, en función de la presencia o ausencia de PPI. Se muestra t-SNE aplicado sobre representaciones aleatorias como referencia.

logística, $p \geq \alpha \Rightarrow$ PPI, $p < \alpha \Rightarrow$ No PPI). El modelo mantiene una precisión alta en un gran rango de valores de sensibilidad, por lo que sus predicciones positivas (presencia de PPI) pueden ser certeras a la vez que se identifican la mayoría de verdaderos positivos. El mejor balance se obtiene con un umbral $\alpha = 0.5$, dando lugar a una precisión de 0.842 y sensibilidad de 0.836. La matriz de confusión muestra la clasificación del conjunto de test utilizando el umbral $\alpha = 0.5$. Como se puede observar, la proporción de falsos positivos y falsos negativos es baja, y la mayoría de pares son clasificados correctamente.

De forma adicional, se decidió visualizar las representaciones individuales agrupadas mediante *global average pooling* utilizando t-SNE (*t-distributed stochastic neighbor embedding*). t-SNE (Maaten & Hinton, 2008) es un método no supervisado y no lineal de reducción de dimensionalidad que refleja las relaciones locales entre puntos de alta dimensionalidad en un plano de dos dimensiones, de forma que los puntos cercanos en el espacio original también lo están en la proyección. Es un método común para la visualización y búsqueda de patrones en representaciones de alta dimensionalidad. Las representaciones individuales de AlphaFold3 para el dataset completo (2500 puntos), agrupadas con *global average pooling* y estandarizadas, se visualizaron en el plano bidimensional utilizando t-SNE con una perplejidad de 30 (provee un balance entre estructura global y local). Como referencia, también se visualizaron representaciones aleatorias de la misma dimensionalidad, cuyos componentes fueron muestreados de una distribución $\mathcal{N}(0, 1)$. Las representaciones de AlphaFold3 se agrupan claramente en función de la presencia o ausencia de PPI, en comparación con las representaciones aleatorias (figura 3.6). Un gran número de representaciones de PPI, forman grupos definidos y disímiles de las representaciones de pares no interactores. Esta visualización muestra un claro patrón en las representaciones individuales de AlphaFold3 cuando los pares de proteínas de entrada interactúan, y explica la eficacia de su uso para la predicción de PPI.

Capítulo 4

Discusión y conclusiones

La predicción de interacciones proteína-proteína es un problema de gran relevancia en el área de la biología molecular y biotecnología. Su resolución tendría un amplio impacto positivo, facilitando el tratamiento de numerosas enfermedades, el desarrollo de fármacos, la optimización de procesos biológicos para la alimentación, entre otros. Resulta particularmente interesante la predicción únicamente a partir de secuencia, ya que es la información más abundantemente disponible. Sin embargo, una gran cantidad de los trabajos realizados hasta el momento han resultado improductivos debido a la presencia de data leakage en los datasets de PPI más utilizados. Los altos rendimientos se transforman en predicciones azarosas cuando se toman medidas para prevenir la filtración de datos de entrenamiento en validación (Bernett *et al.*, 2024). Por tanto, el problema sigue abierto, y es necesario abordarlo con especial cuidado en el diseño de los conjuntos de datos.

En este trabajo se han explorado dos métodos basados en modelos de aprendizaje profundo para la resolución del intrincado problema de predicción de interacciones proteína-proteína. El primero se basó en el uso de ProtBERT, un pLM que ha demostrado tener efectividad en la extracción de información biológica a partir de secuencia. En primer lugar, se replicó la arquitectura ProtBERT-BiGRU-Attention (Gao *et al.*, 2024), recientemente publicada y que afirmaba obtener un alto rendimiento en ausencia de data leakage, en el dataset *gold-standard* de Bernett. Mientras que en su artículo original se logró una accuracy de 0.919, en este trabajo la arquitectura fue completamente incapaz de identificar la presencia de PPI y su rendimiento se mostró indistinguible de la aleatoriedad. Este mismo resultado insatisfactorio se obtuvo con distintas variaciones de los hiperparámetros que no habían sido especificados en el artículo. Aun así, resulta imposible asegurar que la arquitectura es inefectiva, dado que tuvo que ser reimplementada. Esto destaca la importancia de proveer el código utilizado en publicaciones que involucren aprendizaje automático, para asegurar la reproducibilidad y permitir la validación científica de los métodos.

En segundo lugar, se exploró el uso de ProtBERT de forma aislada, con un fine-tuning de todas sus capas en el dataset de Bernett. El resultado también fue una completa falta de aprendizaje, y una clasificación indistinguible de la aleatoria. Esto sugiere que ProtBERT podría ser incapaz de extraer características relevantes para la clasificación de PPI a partir de la secuencia. Siendo este el caso, la efectividad de la arquitectura ProtBERT-BiGRU-Attention, resulta incluso menos plausible, ya que las capas BiGRU y Attention estarían trabajando sobre características no informativas.

Resulta necesario mencionar que el dataset de Bernett tampoco es perfecto para el modelado de PPI. Aunque evita el sobrerrendimiento derivado del data leakage, puede sufrir una falta de evidencia experimental. En su creación, se utilizó la totalidad de las PPI de proteínas humanas contenidas en la base de datos HIPPIE (Alanis-Lobato *et al.*, 2017). Sin embargo, no todos estos pares de proteínas presentan evidencia experimental suficiente para garantizar que existe una interacción física directa. Por tanto, el dataset de Bernett podría contener una proporción significativa de pares positivos que realmente no interaccionan, o al menos, no lo hacen de forma directa. Esta heterogeneidad en el conjunto positivo podría ser un lastre para el aprendizaje de los modelos. Idealmente, el dataset se podría mejorar filtrando aquellas interacciones cuyo respaldo experimental es suficientemente elevado como para garantizar que son directas. Sin embargo, la evidencia experimental a menudo es escasa, y unido a la necesidad de descartar algunos pares para evitar el data leakage, podría dar lugar a un dataset demasiado reducido para el aprendizaje profundo. En el futuro, esta opción debería explorarse para mejorar la calidad de los datasets de PPI.

Por el momento, en este trabajo, se desarrolló un dataset de ratón con alta evidencia experimental y libre de data leakage, a costa de un menor tamaño (solo 2500 pares de secuencias en total). Este dataset se utilizó para validar el uso de AlphaFold3 como extractor de características para la predicción de PPI. Al contrario de lo esperado, las métricas de confianza de interfaz (ipTM y *chain pair PAE min*) no mostraron asociación alguna con la presencia de una interacción real entre las proteínas de entrada. En el artículo de AlphaFold3 estas métricas eran predictivas de la calidad de la estructura del complejo generada para pares de proteínas que sí interaccionaban (Abramson *et al.*, 2024). Sin embargo, esto no debe confundirse con que presenten capacidad para predecir si dos proteínas interactúan o no, afirmación que los resultados de este trabajo sugieren que no es cierta. Por otro lado, las representaciones individuales por tokens generadas por AlphaFold3, sí demostraron una clara capacidad predictiva de PPI. La visualización mediante t-SNE, mostró una notable similitud entre las representaciones que provenían de pares de proteínas interactores, y diferencias frente aquellos pares de proteínas no interactores. Además, una simple regresión logística sobre estas representaciones logró una clasificación con una accuracy alta de 0.840 en un conjunto balanceado, sin recurrir al beneficio del data leakage.

Este trabajo ha demostrado el potencial de AlphaFold3 para predecir interacciones proteína-proteína a partir de secuencia. El método que se ha desarrollado resulta fiable gracias a la ausencia de data leakage y la alta evidencia experimental del dataset empleado, y además presenta un buen rendimiento. Aun así, una de sus principales limitaciones es el largo tiempo de inferencia necesario para predecir cada interacción. El método requiere el costoso cómputo del MSA y *templates* de cada par de proteínas, además del tiempo de inferencia de AlphaFold3, que es varias veces más lento que un pLM como ProtBERT. Esto descarta por el momento su uso en la predicción masiva para completar redes de PPI, aunque puede resultar útil en estudios aislados de unas pocas proteínas.

El éxito de AlphaFold3 es probablemente debido a su alto entendimiento de la relación secuencia-estructura, particularmente en complejos multiméricos. Su preentrenamiento ha forzado desarrollar la capacidad de extraer patrones que permitiesen determinar qué cadenas proteicas interactúan físicamente, a fin de establecer la estructura más plausible de los complejos. El aprendizaje profundo en bioinformática avanza a gran velocidad, y en el último año, se han desarrollado nuevos modelos que

declaran un rendimiento similar a AlphaFold3 en el modelado de complejos multiméricos: Boltz (Passaro *et al.*, 2025) y Chai-1 (Discovery *et al.*, 2024). La última versión de Boltz (Boltz-2), lanzada durante la realización de este trabajo, afirma incluso superar a AlphaFold3 en la predicción de complejos proteína-ligando. Los futuros trabajos para la predicción de PPI se podrían enfocar en aprovechar el conocimiento de AlphaFold3 y estos nuevos modelos, y minimizar el tiempo de inferencia necesario para extraer características suficientes para la predicción de PPI. Aunque se necesite un largo procesamiento para lograr estructuras de calidad comparable a la experimental, es bastante posible que determinar si dos cadenas interactúan o no requiera muchos menos pasos en la inferencia.

4.1. Conclusiones

1. La arquitectura ProtBERT-BiGRU-Attention no demuestra ser efectiva en la predicción de interacciones proteína-proteína a partir de secuencia.
2. El fine-tuning de ProtBERT tampoco es una estrategia eficaz para la predicción de interacciones proteína-proteína a partir de secuencia.
3. Las métricas de confianza de interfaz de AlphaFold3, ipTM y *chain pair PAE min*, no se encuentran relacionadas con la presencia o ausencia de interacción real entre las proteínas de entrada.
4. Las representaciones por tokens de AlphaFold3, y particularmente las representaciones individuales, contienen información relevante sobre la presencia o ausencia de interacción. Una clasificación lineal de las representaciones individuales resulta suficiente para predecir interacciones proteína-proteína con una gran precisión y sensibilidad.

Referencias

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., ... Jumper, J. M. (2024). Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, *630*(8016), 493-500. https://doi.org/10.1038/s41586-024-07487-w

Akbarzadeh, S., Coşkun, Ö., & Günçer, B. (2024). Studying protein–protein interactions: Latest and most popular approaches. *Journal of Structural Biology*, *216*(4), 108118. https://doi.org/10.1016/j.jsb.2024.108118

Alanis-Lobato, G., Andrade-Navarro, M. A., & Schaefer, M. H. (2017). HIPPIE v2.0: Enhancing meaningfulness and reliability of protein–protein interaction networks. *Nucleic Acids Research*, *45*, D408-D414. https://doi.org/10.1093/nar/gkw985

Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G. R., Wang, J., Cong, Q., Kinch, L. N., Schaeffer, R. D., Millán, C., Park, H., Adams, C., Glassman, C. R., DeGiovanni, A., Pereira, J. H., Rodrigues, A. V., van Dijk, A. A., Ebrecht, A. C., ... Baker, D. (2021). Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, *373*(6557), 871-876. https://doi.org/10.1126/science.abj8754

Basu, S., & Wallner, B. (2016). DockQ: A quality measure for protein-protein docking models. *PLOS ONE*, *11*(8), e0161879. https://doi.org/10.1371/journal.pone.0161879

Bernett, J., Blumenthal, D. B., & List, M. (2024). Cracking the black box of deep sequence-based protein–protein interaction prediction. *Briefings in Bioinformatics*, *25*(2), bbae076. https://doi.org/10.1093/bib/bbae076

Campbell, N. A., Urry, L. A., Cain, M. L., Wasserman, S. A., Minorsky, P. V., & Orr, R. B. (2021). *Biology: A global approach* (12th edition, global edition). Pearson Education Limited.

Chen, M., Ju, C. J.-T., Zhou, G., Chen, X., Zhang, T., Chang, K.-W., Zaniolo, C., & Wang, W. (2019). Multifaceted protein–protein interaction prediction based on siamese residual RCNN. *Bioinformatics*, *35*(14), i305-i314. https://doi.org/10.1093/bioinformatics/btz328

Cooper, G. M. (2000). Structure and organization of actin filaments. En *The cell: A molecular approach. 2nd edition*. Sinauer Associates. Consultado el 4 de junio de 2025, desde https://www.ncbi.nlm.nih.gov/books/NBK9908/

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, 24 de mayo). BERT: Pre-training of deep bidirectional transformers for language understanding. https://doi.org/10.48550/arXiv.1810.04805

Dill, K. A., Ozkan, S. B., Shell, M. S., & Weikl, T. R. (2008). The protein folding problem. *Annu Rev Biophys*, *37*, 289-316. https://doi.org/10.1146/annurev.biophys.37.092707.153558

Discovery, C., Boitreaud, J., Dent, J., McPartlon, M., Meier, J., Reis, V., Rogozhnikov, A., & Wu, K. (2024, 15 de octubre). Chai-1: Decoding the molecular interactions of life. https://doi.org/10.1101/2024.10.10.615955

Eddy, S. R. (2009). A new generation of homology search tools based on probabilistic inference. *Genome Inform*, *23*(1), 205-211.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., & Rost, B. (2022). ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE Trans Pattern Anal Mach Intell*, *44*(10), 7112-7127. https://doi.org/10.1109/TPAMI.2021.3095381

Evans, R., O'Neill, M., Pritzel, A., Antropova, N., Senior, A., Green, T., Žídek, A., Bates, R., Blackwell, S., Yim, J., Ronneberger, O., Bodenstein, S., Zielinski, M., Bridgland, A., Potapenko, A., Cowie, A., Tunyasuvunakool, K., Jain, R., Clancy, E., ... Hassabis, D. (2022, 10 de marzo). Protein complex prediction with AlphaFold-multimer. https://doi.org/10.1101/2021.10.04.463034

Ferruz, N., & Höcker, B. (2022). Controllable protein design with language models. *Nat Mach Intell*, *4*(6), 521-532. https://doi.org/10.1038/s42256-022-00499-z

Fiser, A. (2010). Template-based protein structure modeling. En D. Fenyö (Ed.), *Computational biology* (pp. 73-94). Humana Press. https://doi.org/10.1007/978-1-60761-842-3_6

Gao, Q., Zhang, C., Li, M., & Yu, T. (2024). Protein-protein interaction prediction model based on ProtBert-BiGRU-attention. *J Comput Biol*, *31*(9), 797-814. https://doi.org/10.1089/cmb.2023.0297

Guo, Y., Yu, L., Wen, Z., & Li, M. (2008). Using support vector machine combined with auto covariance to predict protein–protein interactions from protein sequences. *Nucleic Acids Research*, *36*(9), 3025-3030. https://doi.org/10.1093/nar/gkn159

Hsu, C., Verkuil, R., Liu, J., Lin, Z., Hie, B., Sercu, T., Lerer, A., & Rives, A. (2022, 6 de septiembre). Learning inverse folding from millions of predicted structures. https://doi.org/10.1101/2022.04.10.487779

Huang, Y.-A., You, Z.-H., Gao, X., Wong, L., & Wang, L. (2015). Using weighted sparse representation model combined with discrete cosine transformation to predict protein-protein interactions from protein sequence. *BioMed Research International*, *2015*(1), 902198. https://doi.org/10.1155/2015/902198

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., ... Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), 583-589. https://doi.org/10.1038/s41586-021-03819-2

Kapoor, S., & Narayanan, A. (2022, 14 de julio). Leakage and the reproducibility crisis in ML-based science. https://doi.org/10.48550/arXiv.2207.07048

Ko, J., & Lee, J. (2021, 27 de julio). Can AlphaFold2 predict protein-peptide complex structures accurately? https://doi.org/10.1101/2021.07.27.453972

Kozakov, D., Hall, D. R., Xia, B., Porter, K. A., Padhorny, D., Yueh, C., Beglov, D., & Vajda, S. (2017). The ClusPro web server for protein–protein docking. *Nat Protoc*, *12*(2), 255-278. https://doi.org/10.1038/nprot.2016.169

Lawrence, M. C., & Colman, P. M. (1993). Shape complementarity at protein/protein interfaces. *Journal of Molecular Biology*, *234*(4), 946-950. https://doi.org/10.1006/jmbi.1993.1648

Lee, J., Freddolino, P. L., & Zhang, Y. (2017). Ab initio protein structure prediction. En D. J. Rigden (Ed.), *From protein structure to function with bioinformatics* (pp. 3-35). Springer Netherlands. https://doi.org/10.1007/978-94-024-1069-3_1

Li, Y., & Ilie, L. (2017). SPRINT: Ultrafast protein-protein interaction prediction of the entire human interactome. *BMC Bioinformatics*, *18*(1), 485. https://doi.org/10.1186/s12859-017-1871-x

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., & Rives, A. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, *379*(6637), 1123-1130. https://doi.org/10.1126/science.ade2574

Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(86), 2579-2605.

Mariani, V., Biasini, M., Barbato, A., & Schwede, T. (2013). lDDT: A local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*, *29*(21), 2722-2728. https://doi.org/10.1093/bioinformatics/btt473

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., & Rives, A. (2021, 17 de noviembre). Language models enable zero-shot prediction of the effects of mutations on protein function. https://doi.org/10.1101/2021.07.09.450648

Mirdita, M., Schütze, K., Moriwaki, Y., Heo, L., Ovchinnikov, S., & Steinegger, M. (2022). ColabFold: Making protein folding accessible to all. *Nat Methods*, *19*(6), 679-682. https://doi.org/10.1038/s41592-022-01488-1

Mocăniță, M., Martz, K., & D'Costa, V. M. (2025). Characterizing host-microbe interactions with bacterial effector proteins using proximity-dependent biotin identification (BioID). *Commun Biol*, *8*(1), 597. https://doi.org/10.1038/s42003-025-07950-y

Moult, J., Pedersen, J. T., Judson, R., & Fidelis, K. (1995). A large-scale experiment to assess protein structure prediction methods. *Proteins: Structure, Function, and Bioinformatics*, *23*(3), ii-iv. https://doi.org/10.1002/prot.340230303

OpenAI. (2025, 16 de abril). *Introducing OpenAI o3 and o4-mini*. Consultado el 5 de julio de 2025, desde https://openai.com/index/introducing-o3-and-o4-mini/

Pan, X.-Y., Zhang, Y.-N., & Shen, H.-B. (2010). Large-scale prediction of human protein-protein interactions from amino acid sequence based on latent topic features. *J. Proteome Res.*, *9*(10), 4992-5001. https://doi.org/10.1021/pr100618t

Pang, Y., & Liu, B. (2023). IDP-LM: Prediction of protein intrinsic disorder and disorder functions based on language models. *PLOS Computational Biology*, *19*(11), e1011657. https://doi.org/10.1371/journal.pcbi.1011657

Passaro, S., Corso, G., Wohlwend, J., Reveiz, M., Thaler, S., Somnath, V. R., Getz, N., Portnoi, T., Roy, J., Stark, H., Kwabi-Addo, D., Beaini, D., Jaakkola, T.,

& Barzilay, R. (2025). Boltz-2: Towards accurate and efficient binding affinity prediction. *bioRxiv*. https://doi.org/10.1101/2025.06.14.659707

Ramaraj, T., Angel, T., Dratz, E. A., Jesaitis, A. J., & Mumey, B. (2012). Antigen–antibody interface properties: Composition, residue interactions, and features of 53 non-redundant structures. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics*, *1824*(3), 520-532. https://doi.org/10.1016/j.bbapap.2011.12.007

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., & Fergus, R. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, *118*(15), e2016239118. https://doi.org/10.1073/pnas.2016239118

Schmirler, R., Heinzinger, M., & Rost, B. (2024). Fine-tuning protein language models boosts predictions across diverse tasks. *Nat Commun*, *15*(1), 7407. https://doi.org/10.1038/s41467-024-51844-2

Schroeder, G. N., Pearson, J. S., & Thurston, T. L. M. (2021). Editorial: Bacterial effectors as drivers of human disease: Models, methods, mechanisms. *Front. Cell. Infect. Microbiol.*, *11*. https://doi.org/10.3389/fcimb.2021.708228

Seychell, B. C., & Beck, T. (2021). Molecular basis for protein–protein interactions. *Beilstein J. Org. Chem.*, *17*(1), 1-10. https://doi.org/10.3762/bjoc.17.1

Singh, R., Devkota, K., Sledzieski, S., Berger, B., & Cowen, L. (2022). Topsy-turvy: Integrating a global view into sequence-based PPI prediction. *Bioinformatics*, *38*, i264-i272. https://doi.org/10.1093/bioinformatics/btac258

Sledzieski, S., Singh, R., Cowen, L., & Berger, B. (2021). D-SCRIPT translates genome to phenome with sequence-based, structure-aware, genome-scale predictions of protein-protein interactions. *Cell Syst*, *12*(10), 969-982.e6. https://doi.org/10.1016/j.cels.2021.08.010

Steinegger, M., & Söding, J. (2017). MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat Biotechnol*, *35*(11), 1026-1028. https://doi.org/10.1038/nbt.3988

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023, 2 de agosto). Attention is all you need. https://doi.org/10.48550/arXiv.1706.03762

Virkamäki, A., Ueki, K., & Kahn, C. R. (1999). Protein–protein interaction in insulin signaling and the molecular mechanisms of insulin resistance. *J Clin Invest*, *103*(7), 931-943. https://doi.org/10.1172/JCI6609

Xiao, Y., Zhao, W., Zhang, J., Jin, Y., Zhang, H., Ren, Z., Sun, R., Wang, H., Wan, G., Lu, P., Luo, X., Zhang, Y., Zou, J., Sun, Y., & Wang, W. (2025, 6 de marzo). Protein large language models: A comprehensive survey. https://doi.org/10.48550/arXiv.2502.17504

Yan, Y., Zhang, D., Zhou, P., Li, B., & Huang, S.-Y. (2017). HDOCK: A web server for protein–protein and protein–DNA/RNA docking based on a hybrid strategy. *Nucleic Acids Research*, *45*, W365-W373. https://doi.org/10.1093/nar/gkx407

Yao, Y., Du, X., Diao, Y., & Zhu, H. (2019). An integration of deep learning with feature embedding for protein–protein interaction prediction. *PeerJ*, *7*, e7126. https://doi.org/10.7717/peerj.7126

Zhang, Y., & Skolnick, J. (2004). Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, *57*(4), 702-710. https://doi.org/10.1002/prot.20264