Universidad del País Vasco
Euskal Herriko Unibertsitatea
FARMAZIA FAKULTATEA
FACULTAD DE FARMACIA
BC MATERIALS
BASQUE CENTER FOR MATERIALS, APPLICATIONS & NANOSTRUCTURES
ehu group

UNIVERSIDAD DEL PAÍS VASCO

EUSKAL HERRIKO UNIBERTSITATEA

FACULTAD DE FARMACIA

DESARROLLO DE HERRAMIENTAS PARA LA PREDICCIÓN DE CONTACTOS INTER-PROTEÍNA

TRABAJO DE FIN DE GRADO

GRADO EN FARMACIA

Autor: Marcos Lequerica Mateos

Director: Joseba Bikandi Bikandi (UPV/EHU)

Codirector: Ivan Coluzza (BC Materials)

Año académico: 2021/2022

Índice

Resumen

DEEPCOV es una red neuronal convolucional capaz de predecir estructuras tridimensionales de proteínas con un alto valor de precisión, utilizando datos de covarianza de las secuencias peptídicas fácilmente obtenibles. La red tiene mejores valores de precisión que las técnicas basadas en Análisis de Emparejamiento Directo demostrando que filtra mejor el ruido evolutivo que tanto afecta a estas técnicas cuando son utilizadas sobre proteínas con pocos homólogos conocidos.

En este trabajo se pretende, a través de un entrenamiento adaptado de la red convolucional, conseguir un modelo de DEEPCOV válido para la predicción de estructuras de complejos proteicos.

En primer lugar, se desarrolló una herramienta para la colecta y procesamiento de los datos del entrenamiento y que trabaja de forma automática. La herramienta fue capaz de generar 2973 archivos de forma autónoma y es útil para su futuro uso, siendo posible adaptarla con facilidad.

En segundo lugar, la red fue entrenada con los archivos generados por la herramienta, obteniendo unos valores de precisión de 18’58%, 15’73% y 19’02% en la predicción de top 1, top 5 y top 10 contactos inter-proteína más cercanos respectivamente. El Valor Predictivo Positivo fue de 40’97%. La baja precisión de la red podría ser explicada por el relativamente bajo número de datos utilizados y la generación de ruido extra en el nuevo procedimiento de generación de datos.

Abstract

DEEPCOV is a convolutional neural network able to predict tridimensional structures of proteins with high precision values using covariance data of the peptide sequences, which is easily obtainable. The network has higher precision values than the Direct Coupling Analysis based techniques, proving that it is better at filtering the evolutive noise. This noise affects the Direct Coupling Analysis techniques so much when used over proteins with few homologues known.

This project meant to, by an adapted training of the network, achieve a DEEPCOV model valid for the prediction of protein complexes' structures. It took a number of steps to harness this.

On first place, a tool for automatic data collection and processing was developed. The tool generated 2973 files for the network's training autonomously. The tool is also usable in the future since its behavior is adapted easily.

On second place, the network was trained with the data generated by the tool, obtaining precision values of 18'58%, 15'73% and 19'02% for the prediction of the top1, top5 and top10 closest intra-protein contacts respectively. The positive predictive value was 40'97%. The low prediction values could be explained with the small amount of data used for the network training, or the introduction of some noise generating steps during the data generation.

Abreviaturas

ANN	Red Neuronal Artificial, *Artificial Neural Network*
ATLAS-EDR	*Clúster* EDR del supercomputador ATLAS del Donostia International Physics Centre
b	*Bias*
BP	Propagación hacia atrás, *Back Propagation*
CASP	*Critical Asessment for Protein Structure Prediction*
CNN	Red Neuronal Convolucional, *Convolutional Neural Network*
DEEPCOV	Red Convolucional utilizada para predicción de contactos proteicos
DCA	Análisis de emparejamiento directo, *Direct Coupling Analysis*
DL	Aprendizaje profundo, *Deep Learning*
FP	Propagación hacia delante, *Forward Propagation*
HMM	Modelo Oculto de Markov, *Hidden Markov Model*
L	Pérdida, *Loss*
LR	Ratio de aprendizaje, *Learning Rate*
ML	Aprendizaje automático, *Machine Learning*
MSA	Alineamiento de múltiples secuencias, *Multiple Sequence Alingment.*
ReLU	Rectificador Lineal Unitario, *Rectified Lineal Unit*

1 Introducción

Las proteínas, desde el origen de la vida consisten en una de las macromoléculas necesarias para su existencia tomando parte directa o indirectamente en la mayoría de procesos necesarios para ello (Alberts, 1998) Muchas de las funciones de las proteínas vienen determinadas por la estructura de la proteína (Halabi et al., 2009). Por lo que, dos proteínas con distinta secuencia, pero con una estructura tridimensional similar pueden tener el mismo rol biológico. La estructura tridimensional de las proteínas se mantiene principalmente mediante interacciones entre los aminoácidos que la forman. Cada tipo de aminoácido tendrá sus propiedades fisicoquímicas y será capaz de participar en distintos tipos de interacciones. Además, también afectarán las propiedades fisicoquímicas de los aminoácidos que se encuentren próximos. Cuando un aminoácido de una proteína cuya función es el mantenimiento de la estructura tridimensional de la proteína muta, las interacciones en las que estaría tomando parte este residuo pueden verse alteradas. Esto, en muchos casos, genera una pérdida de estabilidad en la estructura tridimensional de la proteína, la cual puede resultar en una proteína no viable y que, evolutivamente, se perderá. Por tanto, cuando un aminoácido que participa en interacciones que estabilizan la estructura molecular de la proteína muta, genera un estrés evolutivo hacia los aminoácidos que estarían interactuando con él para que muten también y sean sustituidos por residuos capaces de generar una interacción similar a la que existía anteriormente. Mediante este proceso surgen diversas secuencias que forman proteínas con la misma función y que pueden ser agrupadas en familias.

Dado que la función de gran parte de las proteínas viene determinada por la estructura tridimensional que conforma la secuencia peptídica, es de gran interés la caracterización de la estructura tridimensional de las proteínas. Los métodos experimentales para la determinación de estructuras proteicas han sufrido recientemente grandes avances (Bai et al., 2015), siendo habitual la determinación de estructuras mediante cristalografía de rayos X por su precisión y fiabilidad (Maveyraud & Mourey, 2020). No obstante, su velocidad y coste no pueden competir con los del descubrimiento de nuevas secuencias (Dunham et al., 2012; Reuter et al., 2015), por lo que ha sido necesario el desarrollo de métodos computacionales que permitan una predicción precisa de las estructuras.

Los avances en computación que han surgido en las últimas décadas han permitido el desarrollo de métodos computacionales para la predicción de estructuras tridimensionales de proteínas. Las líneas de investigación abiertas relacionadas con estos métodos, desde su origen, han conseguido una gran popularidad existiendo actualmente proyectos internacionales como el *Critical Assessment of Techniques for Protein Structure Prediction* (CASP). El CASP es un experimento bianual a escala global en el que pueden participar grupos de investigación de cualquier país. Tiene como objetivo la evaluación de los últimos métodos para la predicción de estructuras tridimensionales de proteínas a

partir de su secuencia. Los grupos participantes, reciben secuencias peptídicas cuya estructura tridimensional determinada por métodos experimentales aún no ha sido publicada. Los grupos, utilizando sus herramientas desarrolladas con tal fin, enviarán los modelos predichos para tales secuencias y un comité de expertos del CASP evaluará la precisión de cada método comparando las predicciones con la estructura determinada de forma experimental.

1.2 Métodos computacionales basados en covarianza

Los primeros métodos computacionales para la predicción de estructuras terciarias de proteínas se basan en el principio de que, cuando la mutación de un aminoácido de una secuencia compromete la estructura tridimensional de la proteína y, con ella, la función biológica, otros aminoácidos que interactúan con esa proteína tenderán a mutar también para recuperar esa interacción. Así, los homólogos entre proteínas surgen mediante la mutación de un aminoácido y la mutación de los aminoácidos que interactúan con él para generar una interacción equivalente a la anterior (Kortemme et al., 2004). Esto genera conjuntos de mutaciones emparejadas entre proteínas homólogas que aparecen en posiciones en las que los aminoácidos se encuentran interactuando. Debido a esto, pueden observarse patrones de mutaciones emparejadas a la hora de analizar colecciones de secuencias homólogas. Estas mutaciones emparejadas corresponden muchas veces con los aminoácidos que se encuentran físicamente próximos. Por otro lado, las proteínas homólogas tendrán, todas, una estructura similar. Esto permite que la estructura tridimensional de una proteína pueda, en muchos casos, predecirse analizando una familia entera de proteínas y generando una estructura común para todos los miembros. Estas mutaciones emparejadas se buscan en alineamientos de secuencias múltiples (MSA, *Multiple Secuence Analysis*) y los valores asociados se calculan mediante covarianza. Estos métodos se tratan de buscar una relación entre la covarianza de las mutaciones en posiciones concretas de una secuencia a lo largo de una familia de proteínas y su estructura tridimensional (Skerker et al., 2008; White et al., 2007). Estos resultados consiguen valores altos de precisión siempre y cuando se realice sobre familias con muchos miembros. El problema de estos métodos es la gran cantidad de ruido evolutivo que aparece en las familias, el cual, se hace más notorio cuando la familia cuenta con un número de miembros reducido y es imposible de utilizar sobre secuencias que no cuentan con homólogos conocidos. Además, este método no permite diferenciar los emparejamientos directos o indirectos (aquellos que surgen por otros motivos no relacionados con la interacción directa entre los dos aminoácidos) lo cual genera más ruido aún en las predicciones.

1.3 Métodos computacionales basados en DCA

El análisis del emparejamiento directo (DCA, *Direct Coupling Analysis*) engloba a un conjunto de técnicas que surgen como respuesta al problema que crea el uso de cálculos de covarianza para la predicción de contactos. La eliminación del ruido de fondo que generan los métodos basados en

covarianza, es esencial para la predicción precisa de contactos de corta y larga distancia (haciendo referencia a si los aminoácidos en contacto están próximos en la secuencia o no respectivamente). Las aproximaciones del DCA están basadas en propiedades físicas (Weigt et al., 2009), estadística Bayesiana (Burger et al., 2010) o un método más simple derivado de una propiedad de la matriz de covarianza inversa denominado correlación parcial (Friedman et al., 2008). Estos métodos aumentan considerablemente la precisión de las predicciones respecto a la covarianza y realizan un mejor trabajo eliminando el ruido de fondo. No obstante, estos métodos siguen siendo imprecisos para realizar predicciones sobre secuencias que cuentan con pocos homólogos conocidos. Al igual que los métodos basados en covarianza, estos estudios se realizan sobre MSA.

1.3.1 El alineamiento de secuencias múltiples.

Un MSA es un conjunto de secuencias homólogas alineadas. Este alineamiento consiste en el análisis de una colección de secuencias pertenecientes a una misma familia y, mediante modelos estadísticos, calcular qué residuos se mantienen evolutivamente y cuáles no. En concreto, es muy popular considerar la secuencia como un Modelo Oculto de Markov (HMM, *Hidden Markov Model*) en el que la probabilidad de encontrar un aminoácido en una posición concreta dependería del tipo de aminoácido que se encuentre en la posición anterior pero esas probabilidades para cada situación no son conocidas a priori. No es habitual realizar alineamientos de secuencias, utilizando su secuencia completa. Esto se debe a que en la naturaleza muchos fragmentos similares de secuencias aparecen en muchas proteínas y cuentan con una función concreta para mantener la estructura tridimensional. Por tanto, es más habitual en la práctica clasificar únicamente dominios proteicos (que serían aquellas regiones donde aparece una mayor densidad de plegamientos y, por tanto, mayor densidad de interacciones entre los aminoácidos). Una proteína podría, entonces, contar con varios o ningún dominio. En particular PFAM (Finn et al., 2014) es una base de datos que cuenta con los archivos MSA de más de 19.000 familias de dominios. Muchas proteínas cuentan con uno o varios dominios definidos por PFAM y que, cada uno de ellos, sería una secuencia miembro de una familia y todos los miembros de dicha familia serían fragmentos homólogos de distintas proteínas. PFAM realiza los alineamientos de secuencias utilizando HMM. En su base de datos se encuentran dos tipos de archivos relevantes para este proyecto.

Archivos *full*: Estos archivos están formados por una sucesión de secuencias de dominios proteicos, que pertenecen a una misma familia. En estos archivos las secuencias tienen una longitud variable y se encuentran sin alinear.

Archivos *seed*: En estos archivos las secuencias se encuentran alineadas a través de HMMs. Solamente aparecen en el archivo las posiciones en las que los residuos se encuentran conservados evolutivamente y aparece el carácter *gap* ("-") cuando aparece una adición o deleción de aminoácidos. Estos archivos MSA serían aquellos sobre los cuales se realizan los estudios de covarianza o DCA.

1.4 Métodos computacionales basados en Aprendizaje Profundo

En las últimas décadas han surgido grandes avances en el desarrollo y acerca de los métodos basados en aprendizaje automático (ML, *Machine Learning*) así como un aumento en su popularidad. Estos métodos computacionales siguen un esquema básico en el que se pretende que sea un sistema informático el que, basándose en unas observaciones, desarrolle sus propias hipótesis y sean utilizadas para la predicción o clasificación de nuevos sucesos. El rápido desarrollo que se ha dado los últimos años y que sigue con la misma tendencia, ha permitido el desarrollo de muchos modelos distintos de ML, desde modelos relativamente sencillos como regresiones logísticas o árboles de decisión hasta modelos de gran complejidad como máquinas de soporte vectorial o redes neuronales artificiales (ANN, *Artificial Neural Networks*). Concretamente las ANN forman parte de un subconjunto de técnicas del ML denominadas Aprendizaje Profundo (DL, *Deep Learning*). La particularidad del DL es que el propio método es capaz de extraer las características relevantes de un conjunto de datos a la hora de aprender. En el ML tradicional, un científico de datos se encargaría de extraer las características principales de un conjunto de observaciones para entrenar la herramienta, así como elegir la técnica que considera que más se ajusta para la correcta clasificación de las observaciones. El DL ofrece una mucho mayor flexibilidad a la hora de analizar la relevancia de cada característica y una mayor abstracción a la hora de determinarlas (permite una identificación de características ocultas que no son identificables en otros modelos), es el propio sistema el encargado de discernir las características más relevantes de los datos que recibe.

Recientemente, los métodos basados en DL para la predicción de estructuras tridimensionales de secuencias proteicas han adquirido una gran popularidad y han demostrado valores de precisión muy altos en el CASP11 (Wang et al., 2017), 12 (Xu, 2019), 13 (Wu et al., 2020) y 14 (Jumper et al., 2021). Estos métodos se basan en la utilización de distintas estructuras de redes neuronales como herramientas para aprender la relación entre un conjunto de datos relacionados con la proteína y su estructura tridimensional. Estos son actualmente los métodos más precisos de los que se dispone para la predicción de estructuras proteicas (Jumper et al., 2021). A día de hoy, los métodos basados en DL con mejores valores de precisión utilizan de una u otra forma datos obtenidos a través de técnicas de DCA que posteriormente son utilizados para el entrenamiento de las redes. Entre estos métodos, es cada vez más popular el uso de redes neuronales convolucionales (CNN) (Jones & Kandathil, 2018; Liu et al., 2018).

1.4.1 Introducción al Aprendizaje Profundo

Las redes neuronales artificiales (ANN) son sistemas formados por capas interconectadas de neuronas artificiales. Las neuronas artificiales son entidades matemáticas que reciben una serie de valores numéricos, generan con ellos una activación (que será otro valor numérico) y transmiten esa activación a otro grupo de neuronas. Cada conexión entre neuronas tiene asociado un término peso (w,

weight) (*ver imagen 1*). Una neurona de una capa recibirá los valores de activación de cada una de las neuronas de la capa anterior y cada uno de ellos será multiplicado por el término w asociado a cada conexión (este parámetro w es ajustado durante el entrenamiento). Después, la neurona calcula el sumatorio de todas las activaciones de las neuronas de la capa anterior. Cada una de las activaciones es multiplicada por su w asociado, y al total se le sumará un término *bias* (b), que consiste en un número natural que es independiente para cada neurona (también ajustado durante el entrenamiento). A esta suma se le aplicará después una función de activación como una activación sigmoide o un rectificador linear unitario (ReLU, *Rectified Linear Unit*). El resultado de esta operación es la activación de la neurona, que será después enviado a todas las neuronas de la capa siguiente (que cada conexión tendrá también su w asociado). Esto ocurre sobre todas las neuronas de todas las capas hasta llegar a una capa final donde su salida será la predicción. Este proceso es denominado propagación hacia delante (FP, *Forward Propagation*). El entrenamiento de estas redes se realiza, por lo general, con un conjunto de datos que contienen los datos que pretenden que la red utilice para sus predicciones y los datos reales que se espera que la red sea capaz de predecir. El proceso de entrenamiento comienza con una FP que genera una predicción. La predicción es comparada con el dato real mediante una función de pérdidas, que compararía los datos predichos con los reales y daría como resultado un valor pérdida (L, *Loss*). El valor L se propagaría después en sentido contrario por la red en un proceso denominado propagación hacia atrás (BP, *Back Propagation*). En el BP cada neurona reajusta sus parámetros por un proceso denominado gradiente descendiente, en el que los parámetros w y b son ajustados multiplicados por el resultado de la derivada parcial de cada uno de los parámetros respecto al L y multiplicados también por el ratio de aprendizaje (LR, *Learning Rate*) (que es definido antes del entrenamiento, pudiendo también ajustarse durante el entrenamiento). El FP y el BP se realizan para cada uno de los datos del conjunto de entrenamiento (también pueden agruparse varios datos en grupos denominados *batch*, de esta forma varios datos entrarían en la red sin que se actualicen los parámetros y posteriormente se realiza un BP con un valor L que sería el sumatorio de los L de cada uno de los datos del *batch*). Por lo general, la red se entrena varias veces con el mismo conjunto de entrenamientos. El proceso de entrenar la red haciendo pasar todos los datos del entrenamiento se llama época (comúnmente denominada *epoch*).

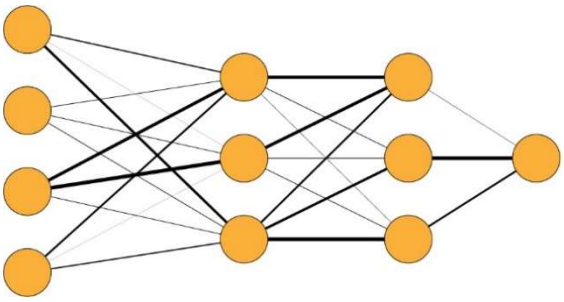


Imagen 1 Las ANN están formadas por varias capas de neuronas en las que, cada neurona de una capa, está conectada con todas las neuronas de la capa anterior y la capa siguiente. Cada una de estas conexiones tiene un peso asociado que multiplica al valor de activación de la neurona anterior de la conexión. El valor peso de cada conexión es ajustado durante el entrenamiento. Elaboración propia.

1.5 La red DEEPCOV

DEEPCOV (Jones & Kandathil, 2018) es una red convolucional desarrollada para la predicción de estructura tridimensional de proteínas. La particularidad de esta red es que, a diferencia de las herramientas basadas en DL más exitosas con este propósito, DEEPCOV utiliza datos mucho más sencillos y accesibles para generar sus predicciones. No utiliza datos derivados de DCA y, sus niveles de precisión son competitivos comparados con los de otras redes que sí los utilizan. Estas redes también utilizan datos de múltiples fuentes, que DEEPCOV demostró que en su mayoría resultan datos redundantes con muy poca información extra que pueda afectar a la calidad de las predicciones. DEEPCOV utiliza datos extraídos del MSA de la secuencia en cuestión únicamente. El hecho de que DEEPCOV consiga obtener cifras de precisión competitivas frente a otras herramientas que utilizan mayor variedad de datos, demuestra que el principal factor para predecir una estructura reside en sus datos de coevolución y que con ellos se pueden explicar gran parte de los contactos y que actúa mejor filtrando el ruido de fondo respecto a las técnicas basadas en DCA o covarianza.

1.5.1 Las redes convolucionales

Las CNN son sistemas especialmente útiles para el reconocimiento de imágenes debido a su capacidad para aprender patrones locales y espaciales (Krizhevsky et al., 2017). Están formadas por capas convolucionales. Las capas convolucionales son capas de neuronas en las que cada neurona se corresponde con un *filtro* que consiste en una matriz o un tensor tridimensional (según el tipo de datos con los que se trabaje). Cada capa puede aprender a reconocer patrones con un grado mayor de abstracción que las capas inmediatamente anteriores a ella.

La red puede tener distintas estructuras: es variable en cuanto al número de capas, número de neuronas en cada capa y parámetros de cada una de ellas e incluso permite combinaciones con otro tipo de capas de neuronas. Todo ello permite que las redes sean diseñadas para cada caso particular para el que vayan a ser utilizadas. La entrada para estas redes está formada por un tensor tridimensional o una matriz (ver imagen 2). Un tensor consistiría en un conjunto tridimensional de datos formadas por matrices de dimensiones similares y un número de canales, que sería el número de matrices que componen el tensor. Cada capa de la red cuenta con un número de *filtros* que, a través de una operación de correlación cruzada entre la matriz/tensor y cada filtro, dan como resultado una matriz con valores altos en las regiones donde el patrón que busca el filtro es similar a la región del tensor sobre la que se está correlacionando y bajos cuando sean muy distintos (*ver imagen 3*). Cada filtro tendrá asociado un término b (similar a las ANN pero en este caso será una matriz de tamaño similar al tensor, pero con únicamente un canal) que se sumará a sus valores y se aplicará una función de activación sobre el resultado.

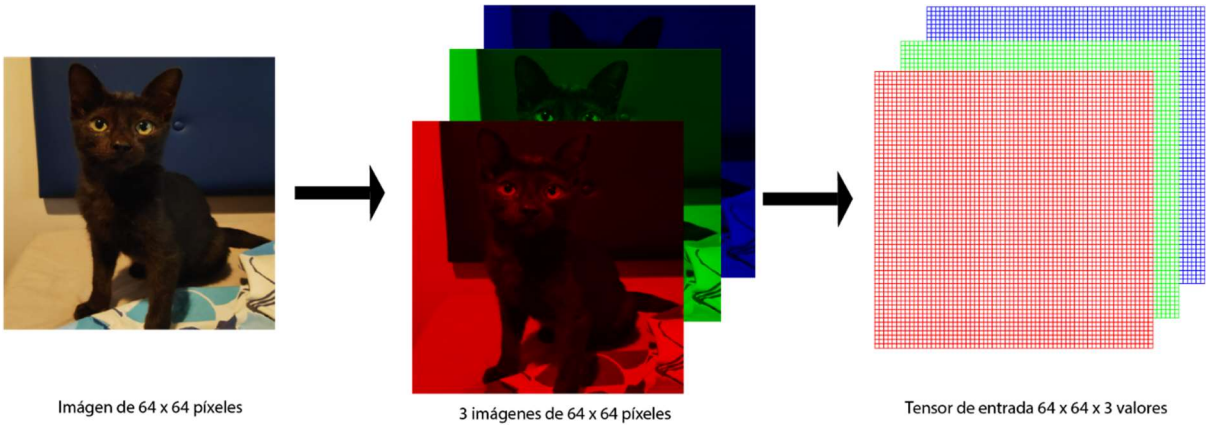


Imagen 2: Un caso habitual es utilizar las CNN para el reconocimiento de imágenes. Una imagen puede ser separada en sus tres canales de color (rojo, verde y azul) y los valores para cada color compondrían una matriz. Después, estas imágenes se agruparían en un tensor tridimensional en las que las dos primeras dimensiones corresponderían al tamaño de las matrices que lo forman (cada una sería un canal) y la última dimensión se correspondería con el número de canales que forma el tensor (en este caso 3 canales). El tensor obtenido, puede ser utilizado como entrada de la CNN. Elaboración propia..

Después, cada una de las matrices obtenidas tras la última operación serán agrupadas en otro tensor tridimensional que pasará a la siguiente capa convolucional como entrada (*ver imagen 4*). Esta segunda capa, realizará el mismo proceso consiguiendo un nivel más de abstracción (p. ej. Una capa puede detectar líneas rectas con distinta orientación con cada filtro y la siguiente capa combinar las activaciones de cada filtro de líneas rectas de la capa anterior para detectar figuras geométricas.). Este proceso es realizado tantas veces como capas haya en la red.

La salida de la capa final de la red consistirá en la predicción realizada por la red (la cual puede ser de distintas formas en función de la intención del proyecto como, por ejemplo, clasificación de imágenes, reconocimiento de objetos en una imagen u otras imágenes). Estas redes aprenden de forma análoga a las ANN tradicionales: 1- Los datos fluyen por la red hasta llegar a una predicción. 2- La predicción es comparada con el valor real mediante la función de pérdidas elegida para calcular un *coste* 3- El valor del *coste* es propagado en sentido contrario por la red y los parámetros (en este caso los *filtros* y el término b) son ajustados mediante *gradiente descendiente* para reducir el *coste* en la siguiente predicción.

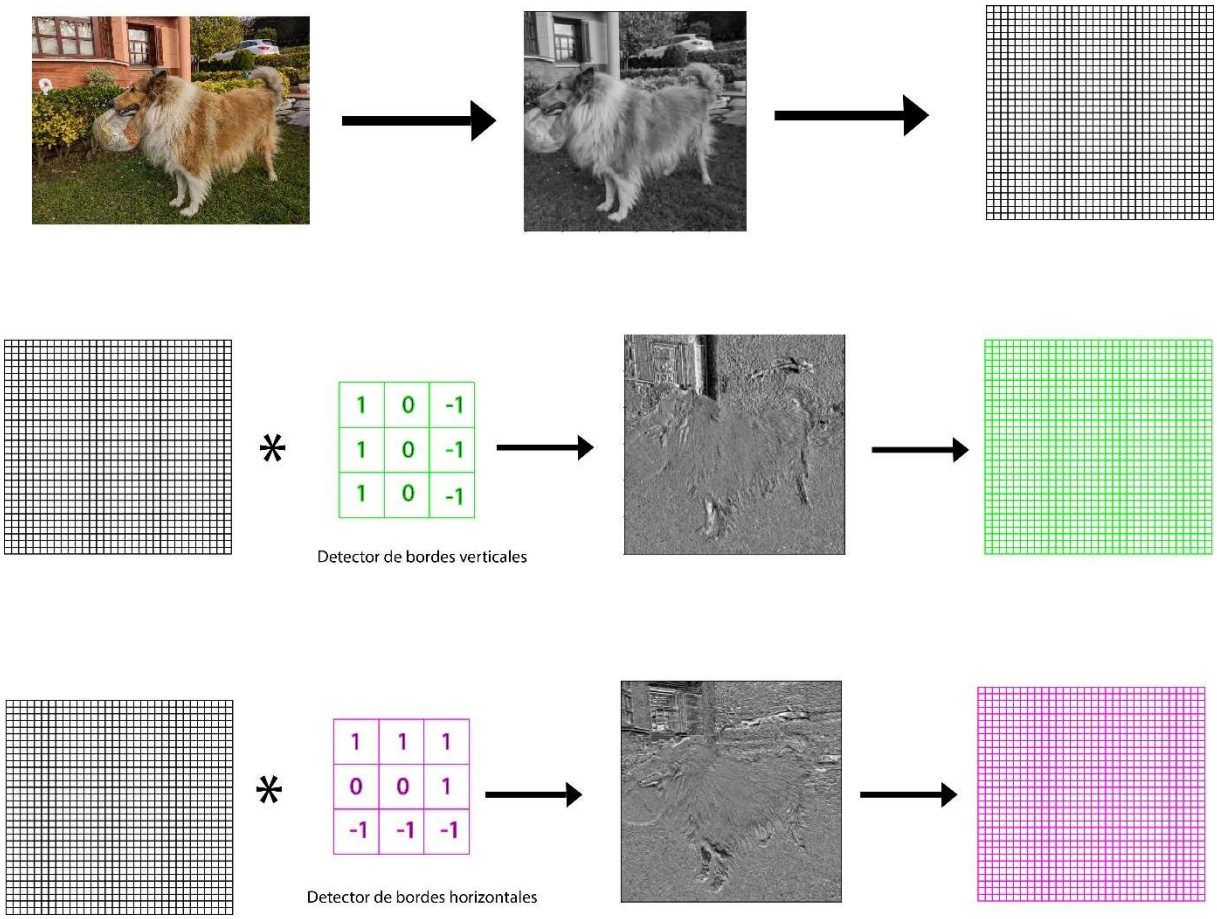


Imagen 3: Ejemplo de operación de correlación cruzada entre una matriz y un filtro. En este caso una foto que ha sido transformada a escala de grises formaría una matriz cuyos valores se corresponderían con la luminosidad de cada pixel. Al realizar la operación entre la imagen y un filtro de bordes verticales (verde) y transformando de nuevo la matriz resultante a una imagen, puede apreciarse valores de mayor intensidad sobre las zonas en las que los valores de la imagen se asemejan con los del filtro (los bordes verticales que aparecen en la imagen). Lo mismo ocurre al realizar la operación con un filtro de bordes horizontales (rosa). Elaboración propia.

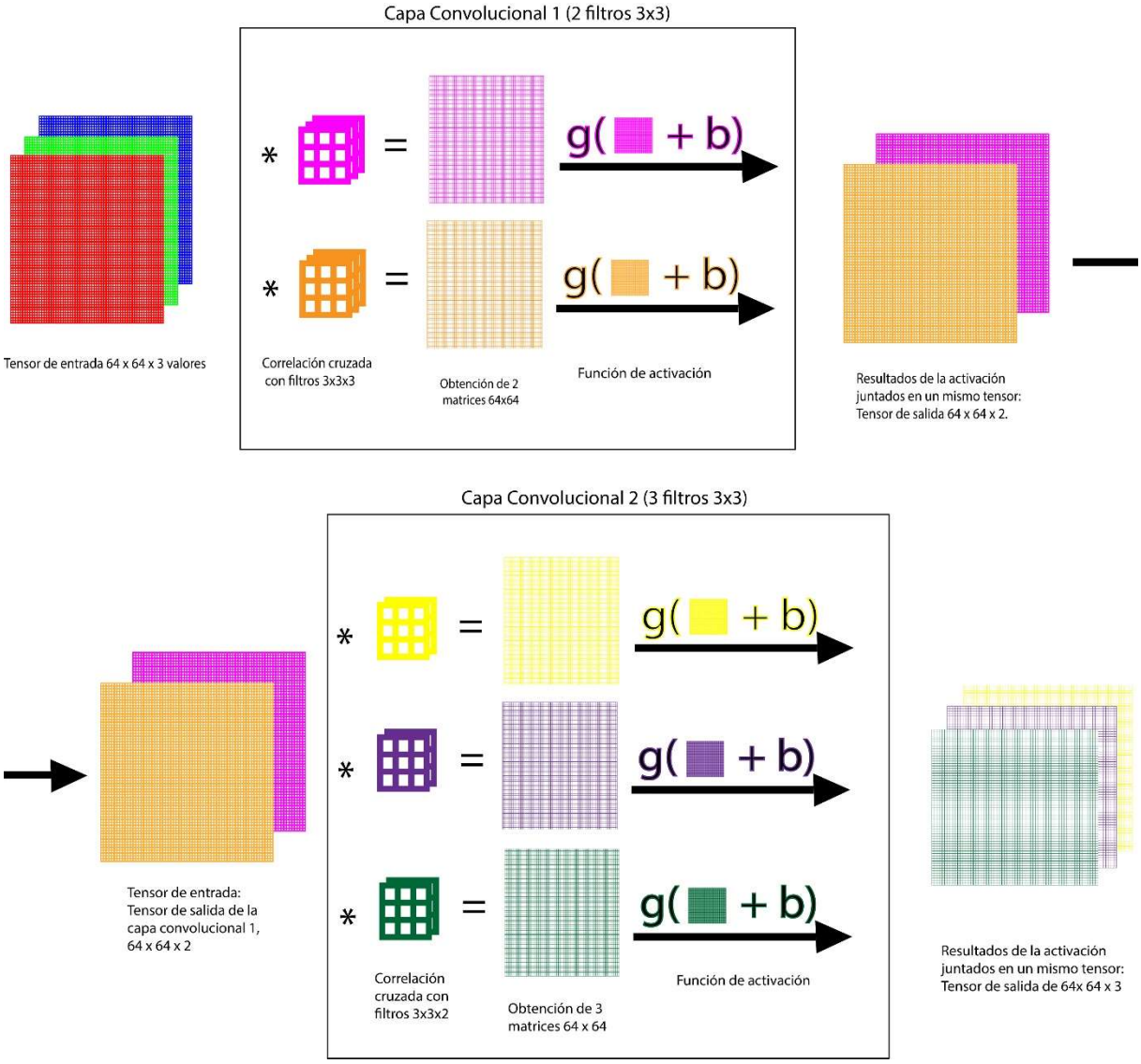


Imagen 4: Cuando un tensor de matrices 64 x 64 con 3 canales entra en una capa convolucional con 2 filtros 3 x 3 (Si la entrada de la capa es un conjunto de datos tridimensional, los filtros también deben serlo. No obstante, solo se menciona el tamaño de las dos primeras dimensiones puesto que, la tercera, tendrá que ser equivalente en tamaño al número de canales del tensor de entrada. En este caso, la tercera dimensión tendrá tamaño 3) en primer lugar se realiza una operación de correlación cruzada (denotada por el término "∗") entre el tensor y cada uno de los filtros (ejemplo en imagen 3). Después, se le suma su término b. Sobre la suma de estos términos se aplica una función de activación. Los resultados de estas operaciones realizadas para cada filtro de la capa, se agrupan en un nuevo tensor que será la entrada de la siguiente capa, que repetirá el mismo proceso. Elaboración propia

1.5.2 Entrada de DEEPCOV

DEEPCOV ha sido probada con dos tipos de datos como entrada. Las *frecuencias de pares* y la covarianza, calculadas sobre los archivos disponibles de MSA para el dominio en cuestión.

Frecuencias de pares: Esta técnica utilizada anteriormente en MetaPSICOV (Jones et al., 2015) consiste en el cálculo de, por cada combinación de dos columnas disponibles en el archivo MSA, la probabilidad de encontrar cada uno de los pares posibles de los 20 aminoácidos canónicos, tratando el *gap* como si se tratara de otro tipo de residuo más. La frecuencia con la que dos sucesos aparecen simultáneamente se definiría como:

$$Freq(X,Y) = E(XY)$$

Donde X e Y consisten en variables binarias que representan observaciones positivas o negativas y $E(XY)$ es equivalente a la probabilidad de encontrar una observación positiva Y, habiéndose observado el suceso X.

Covarianza: La frecuencia con estos dos sucesos ocurren simultáneamente eliminando la contribución de las veces que estos sucesos pueden ocurrir simultáneamente de forma aleatoria dada la concentración en la que dichos sucesos ocurren:

$$Cov(X,Y) = E(XY) - E(X)E(Y)$$

Donde $E(X)$ y $E(Y)$ son equivalentes a la probabilidad de observar los sucesos X e Y de forma independiente. El primer término se referiría a la probabilidad de una observación positiva en Y, habiendo una observación positiva en X (frecuencias de pares). El segundo término eliminaría la contribución directa debida a la concentración de observaciones positivas para ambos casos (se elimina la contribución al cálculo de las asociaciones aleatorias entre sucesos).

En ambos casos los resultados, para una secuencia de longitud n se obtiene una matriz M^{nxn}. Se obtiene una matriz de este tamaño para cada una de las posibles combinaciones entre aminoácidos y/o *gap*. Todos estos datos se agruparán en un tensor de $m\,x\,m\,x\,441$ canales. Estas dimensiones se corresponden con $m\,x\,m$ combinaciones posibles de parejas de posiciones y 441 (21 x 21) combinaciones posibles de parejas de aminoácidos (ver imagen 5).

Los resultados utilizando la matriz de covarianza fueron mejores que los obtenidos utilizando las frecuencias de pares, lo cual demuestra que la red funciona mejor si la contribución de las concentraciones de cada tipo de aminoácido en cada posición concreta en el cálculo es eliminada de antemano.

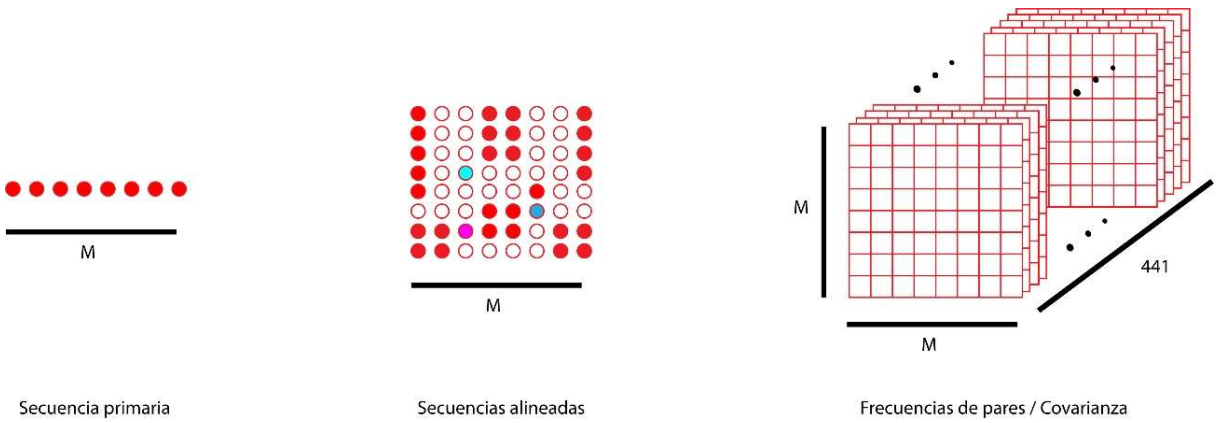


Imagen 5 Una secuencia de una proteína de longitud M es alineada con secuencias homólogas y después se calcula la covarianza/frecuencia de pares. El resultado de esta operación sería un tensor de 441 canales formados por matrices de tamaño M x M. Elaboración propia.

1.5.3 Salida de DEEPCOV

DEEPCOV es utilizada para predecir mapas de contactos. Un mapa de contactos consiste en una matriz con información de qué aminoácidos de una cadena peptídica se encuentran en contacto entre ellos. Es una herramienta útil para la reconstrucción de la estructura tridimensional de una proteína. Para una secuencia de longitud n el mapa de contactos consistirá en una matriz M^{nxn} en las que para cada posición i y j contará con un valor asociado para la existencia de un contacto entre los residuos de posición i y j de la secuencia respectivamente (*ver imagen 6*). La distancia se mide entre los carbonos α (Cα) de dos aminoácidos siguiendo la siguiente fórmula:

$$d(i,j) = \sqrt{(x_i - x_j)^2 + (y_i + y_j)^2 + (z_i + z_j)^2}$$

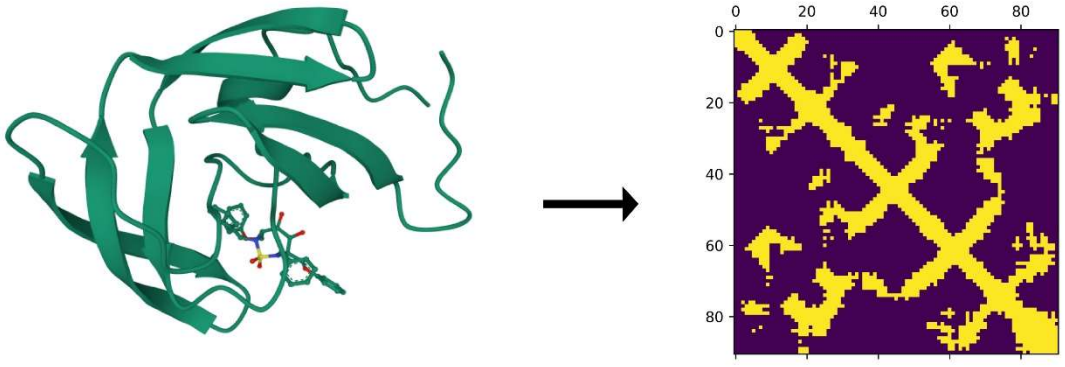


Imagen 6: La estructura general de una proteína puede representarse como un mapa de contactos. El mapa de contactos será una matriz en la que la coordenada de cada eje indica el número de aminoácido al que se refiere. Los valores serán 1 si los aminoácidos que marcan las coordenadas se encuentran en contacto y cero cuando no. Este ejemplo cuenta con la representación tridimensional y el mapa de contactos de un monómero de la Proteasa HIV-1 (Bäckbro et al., 1997) *con código de PDB: 1AJV. La representación 3D está realizada por el software Mol*Viewer* (Sehnal et al., 2021)

Siendo i y j las posiciones en la secuencia de dos aminoácidos y x, y y j las posiciones relativas de los Cα de cada aminoácido. Se considera que dos residuos están en contacto si la distancia entre sus Cα es inferior a 12Å.

1.5.4 Estructura de DEEPCOV

La estructura global de la red (*ver imagen 7*) está formada por múltiples capas. La entrada, el tensor de 441 canales de tamaño $m\ x\ m$ (siendo m la longitud de la proteína), pasa primero por una capa *Maxout* (Goodfellow et al., 2013). Esta primera capa *Maxout* se encarga de reducir el número de canales del tensor, manteniendo sus características principales. De esta forma, el número de datos que contiene se reducen drásticamente perdiendo la mínima información posible. En la red, la capa *Maxout* tiene 64 unidades, por lo que la salida de esta capa será un tensor de 64 canales de matrices mxm. A la capa le suceden un número variable de capas convolucionales de 64 *filtros* de tamaño 3x3 o 5x5. La red ha sido probada con varias estructuras. La capa de salida de DEEPCOV es una capa convolucional con 1 filtro de 1x1 que construiría el mapa de contactos predicho.

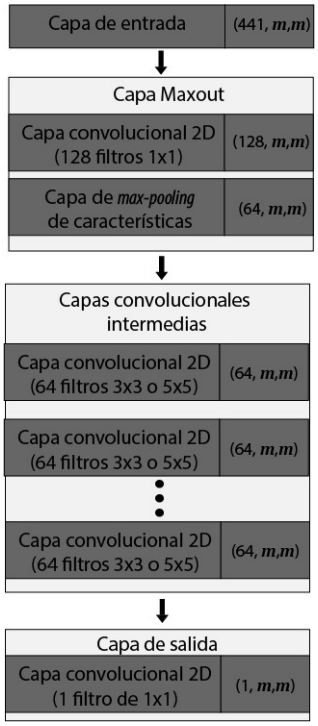


Imagen 7: La estructura de Deepcov se compone por una capa de entrada con la estructura del tensor de entrada (441 x m x m), una capa Maxout de 64 unidades, un número variable de capas convolucionales de 64 filtros y una capa de salida con 1 filtro de tamaño 1 x 1 para completar la predicción. Elaboración propia

1.6 DEEPCOV para predicción de contactos inter-proteína

Las proteínas, en muchas ocasiones, no actúan por si mismas a la hora de desempeñar sus funciones biológicas. Lo hacen formando complejos con otras proteínas. Habitualmente, las dianas biológicas son complejos proteicos o proteínas que los forman para desempeñar una función. El contar con herramientas que predigan con precisión puntos de contacto entre las proteínas que forman un complejo que aún sea desconocido abre las puertas a una investigación más específica a la hora de la búsqueda de nuevas dianas biológicas.

Los métodos basados en DCA son capaces de explicar la mayoría de los contactos que aparecen en una proteína. Estos métodos también permiten predecir puntos de contacto entre proteínas (Lunt et al., 2010; White et al., 2007). Este planteamiento, por tanto, se cumple en casos de interacciones intra e inter-proteína pudiendo explicar muchos de los contactos generados de ambas naturalezas. Puesto que DEEPCOV permite la predicción de puntos de contacto intra-proteína a través de datos mucho más básicos que el resto de sistemas actuales consiguiendo unos valores de precisión aceptables, demuestra que la mayor parte de la información relevante para la predicción de estos contactos se encuentra en estos datos mínimos y que, los diversos

tipos de datos de entrada que usan otras redes (Jumper et al., 2021; Wang et al., 2017; Xu, 2019) consisten, principalmente, en datos redundantes. Esto hace a DEEPCOV una candidata ideal para ser utilizada como marco de desarrollo para una nueva herramienta capaz de predecir puntos de contacto inter-proteína utilizando datos relativamente sencillos.

Además, los avances en las técnicas de *docking* de proteínas (Vreven et al., 2015) reducen la necesidad de altos valores de precisión en cuanto a la predicción de la estructura tridimensional de todo el complejo proteico. Una predicción precisa de únicamente los puntos de contacto más estrechos entre las proteínas del complejo permitiría, en muchos casos, la inferencia de la estructura (del complejo). Esto podría ser realizado utilizando herramientas ya desarrolladas para la predicción de estructuras de proteínas individuales con altos valores de precisión como APHAFOLD 2 (Jumper et al., 2021) y la predicción de zonas de contacto relevantes entre las proteínas como información para el ensayo de *docking*. Dado que las proteínas podrían interactuar en una variedad muy limitada de formas, marcando puntos de contacto, el ensayo de *docking* debería completar una reconstrucción precisa del complejo.

2. Objetivos

Los objetivos para este trabajo experimental fueron los siguientes:

- Crear una herramienta de software que trabaje de forma automática en la creación y procesamiento de datos sobre interacciones proteicas, así como en el almacenamiento de los datos en archivos válidos para ser utilizados en redes neuronales.
- Utilizar la herramienta desarrollada para crear la nueva base de datos con la que será entrenada la red DEEPCOV.
- Validar DEEPCOV para la predicción de contactos inter-proteína.

3. Desarrollo de la herramienta de creación y procesamiento de datos.

3.1 Materiales y métodos

3.1.1 Colecta de datos.

Se desarrolló un script principal (desarrollo propio) escrito en lenguaje de programación Python que automatiza todo el proceso y hace parte del procesamiento. El resto del procesamiento es realizado por scripts auxiliares escritos en C y Bash diseñados por el grupo de Biofísica Computacional de BCMaterials. La etapa final del procesamiento, que consistió en la compresión de los archivos a un formato adaptado se realizó a través de un script (desarrollo propio) escrito en lenguaje Octave.

3.1.1.1 Archivos "pfam"

Los archivos "pfam" (archivos *full* y *seed* explicados en la introducción) pueden ser descargados a través del servicio API de PFAM. No obstante, su funcionamiento a fecha en la que se realizó este proyecto, no era lo suficientemente fiable como para automatizar una colecta de datos que dependiese de él, por lo que se procedió a descargar toda su base de datos, separarla en archivos individuales y transformarlos a formato fasta (el formato que se utilizará posteriormente). Este paso se realizó para tanto los archivos *seed* como para los archivos *full*.

3.1.1.2 Archivos ".pdb"

Los archivos ".pdb" son archivos provistos por la base de datos *Protein Data Bank* (PDB) (Berman et al., 2000) y son largos archivos de texto con información sobre la proteína o complejo proteico del que tratan y la posición relativa de cada átomo de la proteína en el espacio, por lo que en ellos también está definida la estructura. Se filtraron utilizando el buscador de su propia base de datos: archivos de complejos proteicos (tanto homómericos como heteroméricos) cuya estructura haya sido determinada de forma experimental por difracción de rayos X y con una resolución ≤ 2Å. La descarga de los archivos "*.pdb"* fue realizada a través del script que PDB provee para la descarga en masa de archivos. Una modificación propia permitió optimizar su rendimiento permitiéndole descargar los datos de forma simultánea aumentando en gran medida la velocidad. El número total de archivos descargados fue de 33.129.

El script principal (que se ejecutó sobre cada uno de los archivos) descartó todos aquellos que no contaran con al menos 2 cadenas peptídicas con, al menos, 15 residuos de longitud y que en cada una de ellas hubiera, al menos, un dominio de PFAM definido para los que existiera un MSA realizado por esta página (toda esta información está contenida en los archivos ".pdb").

3.1.2 Procesamiento de los datos

Para cada dominio definido en cada una de las cadenas peptídicas del complejo, el primer paso consistió en realizar un MSA entre la secuencia del dominio y las secuencias encontradas en los archivos con los MSA descargados de pfam para esa familia. Este paso fue necesario porque los archivos pfam no cuentan con todas las secuencias conocidas de esa familia, solamente una parte de ellas, por lo que muchas veces la secuencia con la que se trabaja no se encuentra en esa familia. Este paso se realiza utilizando el programa HMMER (Eddy, 1998).

HMMER es un programa utilizado para buscar homólogos de secuencias en bases de datos y para generar alineamientos de secuencias usando HMMs (de forma similar a como PFAM genera sus archivos). En concreto, una de sus utilidades, permite añadir la secuencia en cuestión a los archivos pfam de su familia realizando un MSA entre la secuencia con la que trabaja y las que contienen los archivos pfam. El script auxiliar utilizado para este paso desarrollado por el grupo de Biofísica

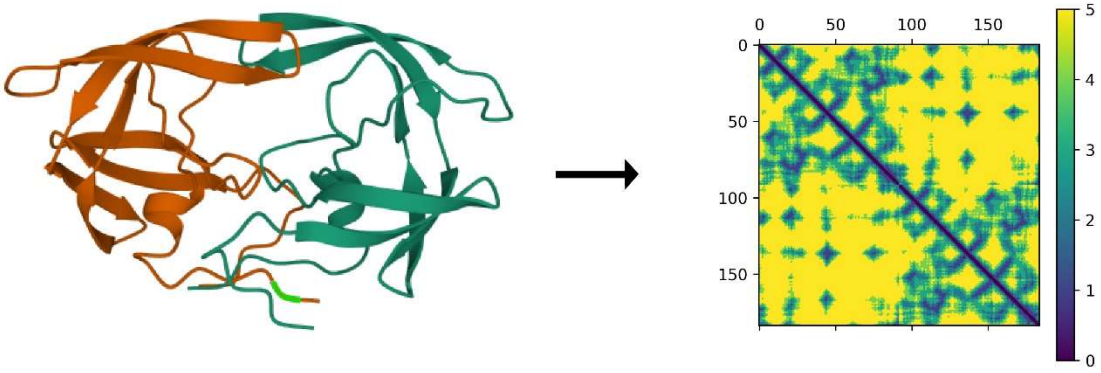
Computacional de BCMaterials y escrito en lenguaje C, se basó en esta utilidad para generar archivos pfam con nuestra secuencia añadida como salida.

Una vez se contaba con la secuencia alineada, se dispuso también de la información de la región de la cadena que se encontraba alineada con el resto de secuencias de la familia (la posición en la que empieza y acaba el dominio), esta región fué sobre la que se trabajó.

Este paso fue realizado sobre cada uno de los dominios descritos por pfam que se encontraban en cada una de las cadenas que formaban cada complejo. Considerando cada combinación posible de dominios que no pertenecieran a una misma cadena como posible interacción inter-proteína, se realizó un mapa de contactos combinado entre los dominios.

> **Mapas de contactos combinados:** Para la creación de un mapa de contactos combinado entre dos secuencias de longitud m_x y m_y respectivamente, se unen las cadenas peptídicas para tratarlas como si fueran una única cadena de longitud m_c. El cálculo de las distancias se realiza de la misma forma que para la creación de un mapa de contactos para una única secuencia. Por tanto, se obtiene una matriz CCM de tamaño $m_c \, x \, m_c$ en las que los valores $CCM_{(i,j)}^{mc \, x \, mc}$ corresponden a la distancia entre residuos que forman parte de cadenas distintas, siempre y cuando ($i > \ m_x$ y $\ j < \ m_x$) o ($j > \ m_x$ y $\ i < \ m_x$).
>
> Para el entrenamiento, a diferencia de en el de DEEPCOV (donde los mapas de contacto eran binarios) los mapas creados fueron categóricos (ver imagen 8a).

El mapa de contactos combinado fue realizado de forma categórica en vez de forma binaria (*ver imagen 8a*). Las categorías para los mapas consistieron en números enteros del 0 al 5 en función de la distancia (*imagen 8b*). El razonamiento tras esta variación en el entrenamiento consiste en que, asignando un valor de importancia a cada contacto, podría conseguirse una mayor especificidad de la red para los contactos más próximos (los únicos realmente necesarios siguiendo la argumentación del punto 1.6) y por tanto una mejor eliminación del ruido.



Categoría	0	1	2	3	4	5
Distancia Å	0 - 4	4 - 8	8 - 12	12 - 16	16 - 20	> 20

Imagen 8. [a] Un mapa de contactos combinado tiene estructura similar a un mapa de contactos convencional. Cada eje representará la numeración del aminoácido al que se refiere pero, en este caso, las dos cadenas se unen y se tratan como si fuera una única cadena. De esta forma para dos cadenas de longitud X e Y la longitud total será X + Y y cada coordenada C hará referencia a un aminoácido de la primera proteína si C < X y a la segunda si C > X. Las regiones de la matriz CCM (C1 > X, C2 < X) y CCM (C1 < X, C2 > Y) consistirán el la región de mapa que contiene los contactos de naturaleza inter-proteína, siendo C1 y C2 las posiciones en la cadena empalmada de los dos aminoácidos en cuestión respectivamente. Este ejemplo cuenta con la representación tridimensional y el mapa de contactos de un monómero de la Proteasa HIV-1 (Bäckbro et al., 1997) *con código de PDB: 1AJV. La representación 3D está realizada por el software Mol*Viewer* (Sehnal et al., 2021)

[b] El mapa de contactos combinado se calculó de forma categórica donde las categorías de 0 a 5 representan distancias entre 0-4Å, 4-8 Å, 8-12 Å, 12-16 Å, 16-20 Å o más de 20Å respectivamente. Elaboración propia.

A esta etapa le sucede una selección de las interacciones apropiadas. En el caso de proteínas heteroméricas, fueron consideradas todas las potenciales interacciones generadas en el paso anterior. En el caso de las homoméricas aparecieron problemas relacionados con: 1. Interacciones redundantes en caso de homo-n-meros con múltiples cadenas. 2. Interacciones no existentes dado que los dominios pertenecían a proteínas que se encontraban físicamente alejadas entre ellas y no existía posibilidad de interacción (ver imagen 9). En estos casos, esa interacción contaría con un mismo archivo MSA (que al final acabarían generando los mismos datos) que dos dominios equivalentes que sí que estuvieran próximos entre ellos, pero con un mapa de contactos combinado vacío en las regiones que corresponderían a las zonas de contacto inter-proteína y, por tanto, no sería apropiado incluirlos en el entrenamiento. El proceso elegido para seleccionar las interacciones que fueron utilizadas (en homómeros) consistió en el cálculo del número de contactos que se detectaron de naturaleza inter-proteína para cada potencial interacción. Después, de cada combinación concreta de dominios, se seleccionaron aquellos en la que aparecían más contactos.

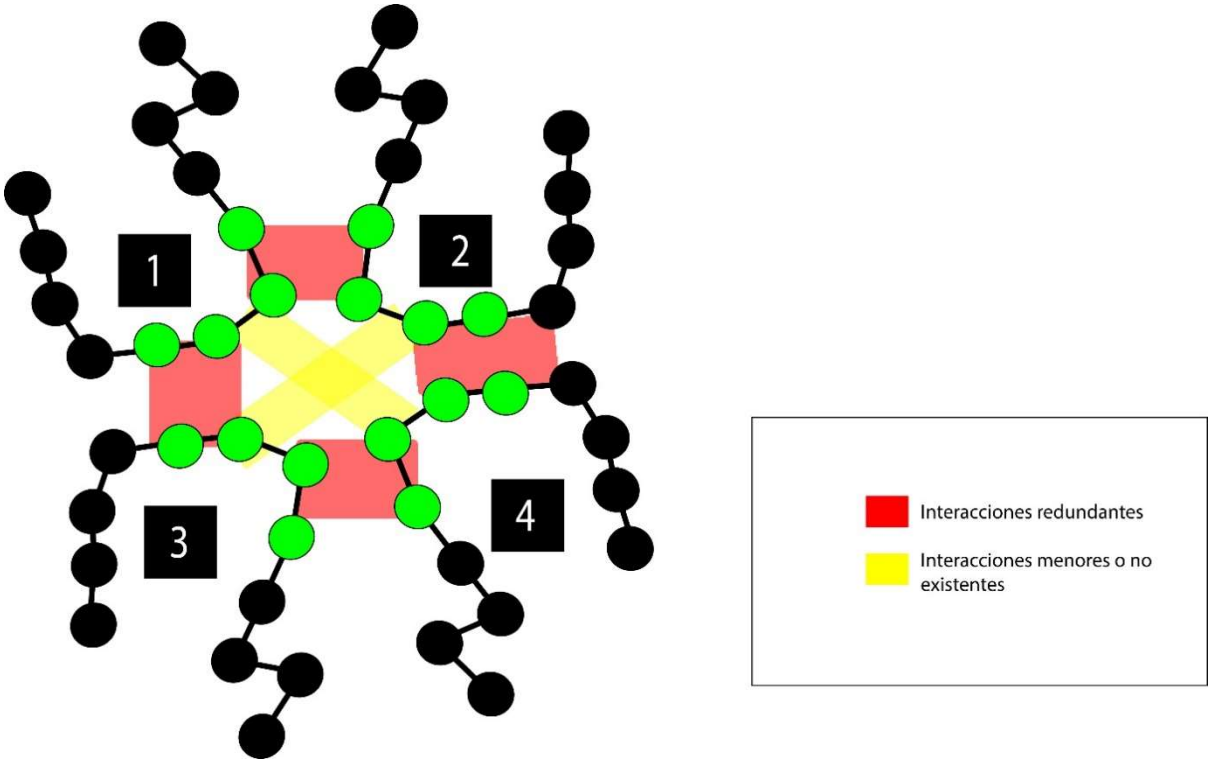


Imagen 9. Este dibujo ilustra el caso de un homo-4-mero en los que los círculos negros representarían aminoácidos que no pertenecen a un dominio pfam descrito y los verdes a los que sí lo hacen. Las regiones rojas consistirían en zonas de interacción entre dominios en las que aparecen las interacciones necesarias para la formación del complejo. Al ser todas las proteínas del complejo iguales, las 4 interacciones de color rojo tendrán el mismo mapa de contactos o uno muy similar. Los 4 archivos que se podrían generar con las interacciones en rojo serían datos redundantes, por ello, el script solamente seleccionará una interacción de entre todas las rojas para su procesamiento. Las regiones en amarillo son interacciones no existentes dado que los aminoácidos de las proteínas en cuestión no interaccionan (o lo hacen en pequeña medida) porque se encuentran físicamente alejadas dentro del complejo, no porque las secuencias como tal no estén destinadas a interactuar. Por este motivo el script descartará este tipo de interacciones para evitar entrenar la red con mapas de contactos vacíos de secuencias que sí que están destinadas a interactuar. Elaboración propia.

Sobre las interacciones seleccionadas se realizó un proceso de combinación de secuencias destinado a generar archivos equivalentes a los archivos MSA de cada familia, pero que en este caso hicieran referencia a una combinación de familias, que son referidos como MSA combinados (MSA_c.) Para ello, se realizó una combinación de secuencias.

Combinación de secuencias: El archivo MSA_c se crea a partir de los dos archivos MSA pertenecientes a los dos dominios que se encuentran interactuando en la proteína. En ellos, se combinarán las secuencias de cada uno de los archivos MSA encadenándolas y siendo tratadas como una única secuencia (ver imagen 10). Las combinaciones se realizaron solamente si los organismos a los que pertenece cada una de la secuencia eran los mismos (suponiendo que estas serían las secuencias que tienen oportunidad de interactuar) y siempre y cuando generaran secuencias con una diferencia mayor que un 30%. Esta etapa se realizó en un script auxiliar escrito en *Bash* por el grupo de Biofísica computacional de BCMaterials.

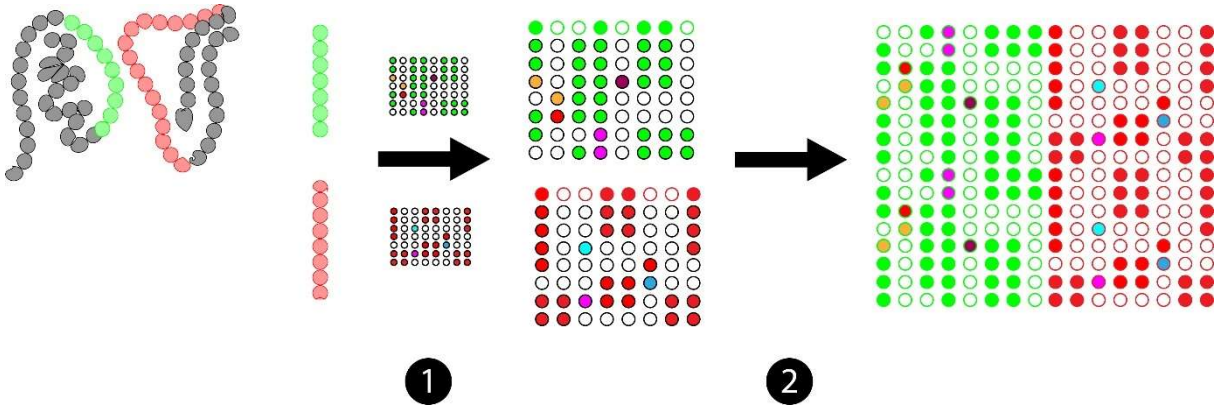


Imagen 10: La secuencia de cada dominio es añadida y alineada con el archivo MSA de la familia a la que pertenece (1). Después, el proceso de combinación de secuencias entre los archivos MSA generaría los archivos MSA combinados (2).

Cálculo de la covarianza: El cálculo de la covarianza se realiza a través de un *script* escrito en lenguaje *C* desarrollado por el grupo de Biofísica computacional de BCMaterials. De la misma forma que en DEEPCOV, se realiza un cálculo de la covarianza entre pares de posiciones y los residuos que aparecen en el lugar para buscar mutaciones emparejadas durante la evolución de la secuencia. El resultado del cálculo consistirá en un tensor de tamaño $21\ x\ 21\ x\ n\ x\ n$ que fue utilizado como entrada de la red.

Archivos H5: Los datos del mapa de contactos y la covarianza fueron almacenados en archivos H5. Este formato de archivos está optimizado para almacenar grandes cantidades de datos multidimensionales de naturaleza numérica de forma jerárquica, lo que los hace idóneos para almacenar este tipo de datos. En ellos se guardaron los cálculos de la covarianza, para la entrada de la red, y los mapas de contacto combinados para el cálculo del *coste* al introducir sus valores y los de la salida de la red en la función de pérdidas.

El procesamiento de los datos fue realizado en el supercomputador Atlas del Donostia International Physics Center, en el *clúster* EDR (ATLAS-EDR).

Fue necesaria una selección de los datos por la longitud total de la combinación de las cadenas de cada uno de los dominios. En la etapa del cálculo de la covarianza, dado que el tensor a generar tenía por

tamaño $m\ x\ m\ x\ 21\ x\ 21$, el espacio de almacenamiento requerido para estos datos, así como el coste computacional para el cálculo de la covarianza, aumentaban de forma exponencial con la longitud de la cadena combinada. Fueron descartados todas las interacciones con una longitud de cadena combinada de más de 250 residuos y todos aquellos cuyo archivo generado con el tensor de covarianza ocupara más de 200MB de almacenamiento. Esta etapa disminuyó en gran medida el número de datos viables para su procesamiento.

3.2 Resultados

A través de la herramienta desarrollada se pudieron obtener un total de 2937 datos, cada uno de ellos con información acerca de una interacción entre dos dominios que pertenecen a proteínas distintas que forman parte de un complejo proteico. El espacio de almacenamiento medio para los datos fue de 53 MB, ocupando el set de datos completo un total de 176,7 GB. Todos los archivos contaron con: 1- Un nombre de fichero que hacía referencia a el código de PDB del complejo, las cadenas en las que aparecía cada dominio, la familia de PFAM a la que pertenecía cada dominio y las coordenadas del origen y el final de cada dominio. 2- El tensor tetradimensional con los datos de covarianza. 3- El mapa de contactos combinado categórico.

4 Entrenamiento de la red

4.1 Materiales y métodos.

Para el entrenamiento de la red fueron utilizados todos los 2973 archivos generados por la herramienta de colecta de datos. De ellos, un 90% se utilizó para el entrenamiento de la red y un 10% para las pruebas y validación. Se utilizó una versión propia de un script desarrollado por la empresa Acuratio que recrea la estructura original de DEEPCOV definida por Jones y Kandathil en 2018, utilizando como marco de trabajo Keras con el *backend* de Tensorflow. El entrenamiento fue realizado de dos formas distintas, una de ellas en un ordenador portátil y la otra en ATLAS-EDR. Ambas siguieron el esquema general de DEEPCOV (*ver imagen 4*). La red entrenada de forma local se diseñó con 9 capas convolucionales con filtros de tamaño 3x3. La red entrenada en ATLAS-EDR se diseñó con 12 capas convolucionales con filtros de tamaño 5x5. En ambos casos se modificó la capa de salida, contando al final con 6 filtros de tamaño 1x1 donde la activación de cada uno de los 6 filtros (uno por cada categoría posible, *ver imagen 6*) corresponden con la probabilidad que la red predice de que un punto del mapa de contactos se encuentre en esa categoría. La separación de los datos se realizó de forma automática y aleatoria en los primeros pasos del script, por lo que cada una de las redes utilizó datos distintos para su entrenamiento. Los siguientes parámetros fueron similares en ambos entrenamientos:

1. El optimizador utilizado fue Adamax (Kingma & Lei Ba, 2014) con un valor de 0’002.
2. La función de pérdidas utilizada fue la entropía cruzada categórica. Cabe destacar que las categorías en referencia a los contactos son categorías infrarrepresentadas y el sistema podría

beneficiarse utilizando funciones de pérdidas adaptadas para equilibrar la importancia de las clases infrarrepresentadas en las predicciones como *Focal loss* (Lin et al., 2017) .

3. Se añadieron parámetros para evitar *overfitting*:
    a. Se definió un *early stopping* monitorizando el coste y con una paciencia de 10 (Por lo tanto, si el valor del coste medio para el conjunto de datos completo no cambia durante más de 10 *epochs* la red terminaría automáticamente el entrenamiento aun no habiendo llegado al número de *epochs* definido inicialmente).
    b. Se definió un *reduce learning rate on plateau* de 0.5 y paciencia de 5 monitorizando, igualmente, el coste (Si el coste no cambiara durante más de 5 *epochs*, el ratio de aprendizaje se reduciría a la mitad).
4. Se definió un tamaño de *batch* de 1, para reducir el uso de memoria.

El número de *epochs* definido fue 100 para el entrenamiento local y de 300 para el de ATLAS-EDR. Ambos entrenamientos completaron el número de *epochs* definidos sin que saltara el *early stopping*. El entrenamiento local se realizó en la tarjeta gráfica *Nvidia GTX 1050 mobile* con 4GB de VRAM en un ordenador portátil con 20GB de RAM. El espacio de almacenamiento que ocupaba todo el conjunto de datos fue de 176,7GB que fue almacenado en un disco duro NMVE del mismo portátil. Este paso fue crucial para el entrenamiento local ya que, al utilizarse discos duros más lentos para almacenar los datos, se observó que la velocidad de entrenamiento de la red se veía gravemente afectada debido al tiempo de lectura de cada archivo. El tiempo total de entrenamiento de la red fue de 71 horas para el entrenamiento local y se realizó de forma continua. El entrenamiento en el *cluster* se realizó sobre un nodo con 64gb de memoria RAM y una tarjeta gráfica RTX 3090 de 24 GB de VRAM y tardó un total de 125h de entrenamiento.

4.2 Resultados

El objetivo del uso de esta red como herramienta auxiliar en ensayos de docking permite que sea de gran utilidad siendo capaz de predecir solamente ciertos contactos clave (ya que serviría como referencia principal para el docking, a través del cual se podrían predecir el resto de los contactos no predichos por la red solamente por pura geometría de cada una de las proteínas). Debido a esto, los criterios de evaluación seguidos fueron los siguientes:

- Precisión en la predicción de *topN* contactos: Se evaluó el porcentaje promedio de los *N* contactos más próximos de cada caso que fueron predichos por la red. Los números elegidos fueron *N = 1, 5, 10*.
- Valor Predictivo Positivo (VPP): La probabilidad de que un contacto predicho por la red sea realmente un contacto.

$$VPP = \frac{Verdaderos\ Positivos}{Verdaderos\ Positivos + Falsos\ Positivos}$$

Ambos valores se calcularon únicamente sobre los contactos de naturaleza inter-proteína a través de un *script* de desarrollo propio y se calcularon sobre el conjunto de entrenamiento y de pruebas respectivamente para poder detectar un posible caso de *overfitting*. En ambos casos se dio como válido la predicción de cualquier categoría inferior a 4 para un contacto como válida.

En la *Tabla 1* quedan recogidos los resultados obtenidos de cada una de las redes para sus respectivos conjuntos de datos de entrenamiento y de validación.

	Top 1 (%)	Top 5 (%)	Top 10 (%)	VPP (%)
Pruebas A	4’25	6’25	9’32	36’39
Entrenamiento A	4’18	6’87	9’79	38’09
Pruebas L	18’58	15’73	19’02	40’97
Entrenamiento L	20’6	18’51	20’81	40’18

Tabla 1: Valores obtenidos para el cálculo de la precisión en Top1, 5 y 10 contactos y VPP en los conjuntos de Entrenamiento y Pruebas en la red entrenada en ATLAS-EDR(A) y la red entrenada de forma local (L).

En la *Imagen 11* queda representado un ejemplo de las predicciones que realiza cada red para una misma interacción que perteneció ambos conjuntos de entrenamientos.

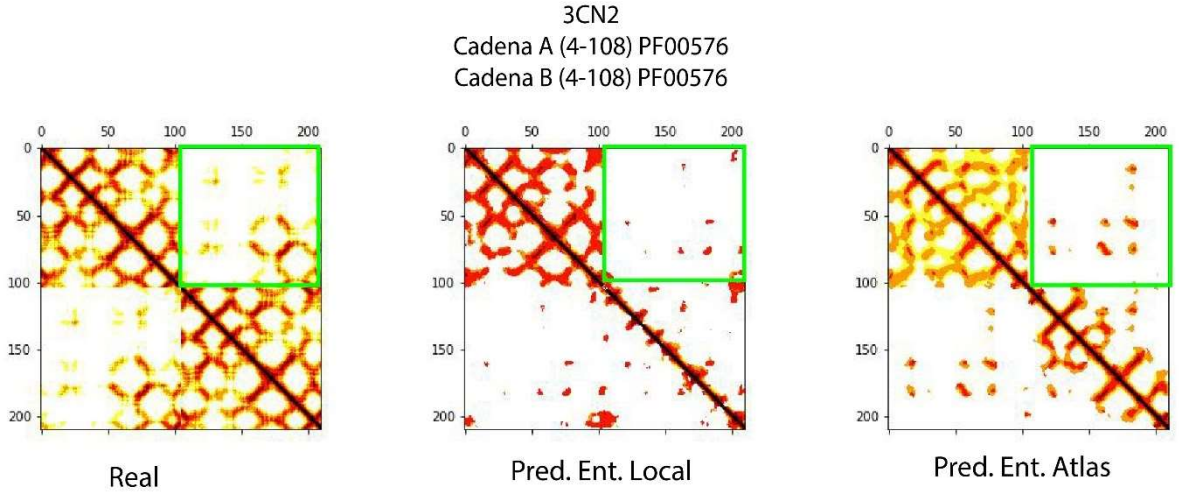


Imagen 11: La región representada en verde (sector 2) es la valorada en las pruebas de precisión. Este caso aquí presentado muestra las mayores diferencias apreciables entre los mapas de contactos predichos por las dos redes. Puede observarse que la red entrenada en Atlas predice un mayor número de contactos tanto intra como inter-proteína (que, según los datos obtenidos, tienen un mayor porcentaje de falsos positivos). En este caso puede apreciarse claramente que la predicción para el sector 1 del mapa de contactos es muy precisa y no lo es la del sector 3 (La proteína en cuestión es un homómero y en el mapa real puede observarse que ambas regiones son idénticas) así como que este fenómeno es más notorio en la red entrenada de forma local.

5. Conclusiones

La herramienta de creación y procesamiento de archivos cumplió correctamente con su función de crear archivos listos para el entrenamiento de la red a partir de un archivo PDB. Permite, también, modificar el procesamiento de los datos a través de la modificación de los *scripts* auxiliares y/o el principal de forma sencilla. No obstante, para un uso viable en la creación de archivos de gran tamaño sería apropiada una optimización del programa.

La red neuronal presenta valores de precisión bajos en cuanto a la predicción de puntos de contacto inter-proteína, no siendo actualmente apropiada para ningún uso real. Los mejores resultados obtenidos con la red más sencilla se deben a que este diseño predice un número menor de contactos dando mayor relevancia a los contactos más cercanos y, también, un número menor de falsos positivos (ver ejemplo en imagen 9). Los mapas de contactos, por lo general, presentaban una gran diferencia entre los sectores 1 y 3 del mapa (regiones correspondientes a contactos intra-proteína), siendo las predicciones precisas, en muchos casos, para el sector 1 y muy imprecisas para el sector 3 (tanto en homómeros, donde ambas regiones deberían ser simétricas, como en heterómeros). Este fenómeno es más apreciable en el modelo de red más sencillo (aquella entrenada de forma local). Esto puede deberse a una generación asimétrica de los alineamientos múltiples en el procesamiento de los datos (ver sección 3.1.2). Este paso, sin duda, genera una gran cantidad de ruido que podría estar afectando gravemente a los resultados reales. Con este procedimiento, para generar los el MSA combinado para dos familias, se combinan todas las secuencias de cada una de las familias que pertenezcan al mismo organismo. Este paso solamente podría ser válido si todas las secuencias combinadas siguiendo este criterio estuvieran destinadas a interactuar entre ellas formando complejos, hecho que no se da. El uso de una herramienta como ACT-SVM (Ma et al., 2020) (Una herramienta basada en una Máquina de Soporte Vectorial para la predicción de si dos secuencias estarán destinadas a interactuar que ha conseguido altos valores de precisión para algunos organismos) durante este paso para filtrar gran parte de las combinaciones secuencias que no interactúan podría mejorar los resultados del proyecto. El uso de una función de pérdidas que priorice la clasificación correcta de las categorías menos representadas podría evitar la tendencia a la predicción de la categoría 5 (distancia mayor a 20Å, categoría más presente en los mapas de contactos). Podría utilizarse una función como *focal loss* o aplicar pesos a las categorías en función de su aparición. También sería de interés el diseño de una función de pérdidas que solamente calculara el coste de la región inter-proteína. Las diferencias entre los valores de precisión para los conjuntos de entrenamiento y los de test no muestran, a primera vista, signos de un excesivo *overfitting.* Aun así, esto debería ser comprobado en futuros entrenamientos de la red a través de métodos de validación como el *k-fold Cross Validation* (método que consiste en la separación del conjunto de datos en *folds* y se entrenarán tantos modelos como *folds* hayan sido definidos. Cada modelo utilizará un *fold* distinto como conjunto de pruebas y el resto como conjunto de entrenamiento. Si no existiera *overfitting* los valores de precisión para cada modelo deberían ser

similares al resto). Por otro lado, el número de datos utilizados es bajo. El gran espacio de almacenamiento requerido, el coste computacional necesario para la producción de datos y problemas relacionados con ATLAS-EDR limitaron enormemente el número de datos que fue posible procesar. Si bien el script principal fue preparado para filtrar y adaptarse a irregularidades en los archivos ".pdb", muchas de ellas no eran contempladas (impidiendo procesar dichos archivos y categorizándolos como erróneos) aún y podrá ser adaptado a ellas fácilmente en el futuro para una obtención de un mayor número de archivos. Un número mayor de datos podría aumentar la precisión de la red (pruebas preliminares con pequeños subconjuntos de datos mostraban una precisión muy inferior a la red definitiva). Por tanto, también sería de interés una colecta de un mayor número de datos para pruebas futuras.

4 Bibliografía

Alberts B. (1998). The cell as a collection of protein machines: preparing the next generation of molecular biologists. *Cell*, *92*(3), 291–294. https://doi.org/10.1016/s0092-8674(00)80922-8

Bäckbro, K., Löwgren, S., Osterlund, K., Atepo, J., Unge, T., Hultén, J., Bonham, N. M., Schaal, W., Karlén, A., & Hallberg, A. (1997). Unexpected binding mode of a cyclic sulfamide HIV-1 protease inhibitor. *Journal of medicinal chemistry*, *40*(6), 898–902. https://doi.org/10.1021/jm960588d

Bai, X. C., McMullan, G., & Scheres, S. H. (2015). How cryo-EM is revolutionizing structural biology. *Trends in biochemical sciences*, *40*(1), 49–57. https://doi.org/10.1016/j.tibs.2014.10.005

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., & Bourne, P. E. (2000). The Protein Data Bank. *Nucleic acids research*, *28*(1), 235–242. https://doi.org/10.1093/nar/28.1.235

Burger, L., & van Nimwegen, E. (2010). Disentangling direct from indirect co-evolution of residues in protein alignments. *PLoS computational biology*, *6*(1), e1000633. https://doi.org/10.1371/journal.pcbi.1000633

ENCODE Project Consortium (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, *489*(7414), 57–74. https://doi.org/10.1038/nature11247Eddy, S. R. (1998). Profile hidden Markov models. *BIOINFORMATICS REVIEW*, *14*(9), 755–763.

Finn, R. D., Bateman, A., Clements, J., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Heger, A., Hetherington, K., Holm, L., Mistry, J., Sonnhammer, E. L., Tate, J., & Punta, M. (2014). Pfam: the protein families database. *Nucleic acids research*, *42*(Database issue), D222–D230. https://doi.org/10.1093/nar/gkt1223

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics (Oxford, England)*, *9*(3), 432–441. https://doi.org/10.1093/biostatistics/kxm045

Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A. & Bengio, Y.. (2013). Maxout Networks. Proceedings of the 30th International Conference on Machine Learning, in Proceedings of Machine Learning Research 28(3):1319-1327 Available from https://proceedings.mlr.press/v28/goodfellow13.html.

Halabi, N., Rivoire, O., Leibler, S., & Ranganathan, R. (2009). Protein sectors: evolutionary units of three-dimensional structure. *Cell*, *138*(4), 774–786. https://doi.org/10.1016/j.cell.2009.07.038Jones, D. T., & Kandathil, S. M. (2018). High precision in protein contact prediction using fully convolutional neural networks and minimal sequence features. *Bioinformatics*, *34*(19), 3308–3315. https://doi.org/10.1093/bioinformatics/bty341

Jones, D. T., Singh, T., Kosciolek, T., & Tetchner, S. (2015). MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. *Bioinformatics (Oxford, England)*, *31*(7), 999–1006. https://doi.org/10.1093/bioinformatics/btu791

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., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., … Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2Kingma, D. P., & Lei Ba, J. (2014). *ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION*. . . ArXiv Preprint ArXiv:1412.6980.

Kortemme, T., Joachimiak, L. A., Bullock, A. N., Schuler, A. D., Stoddard, B. L., & Baker, D. (2004). Computational redesign of protein-protein interaction specificity. *Nature structural & molecular biology*, *11*(4), 371–379. https://doi.org/10.1038/nsmb749

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. *Communications of the ACM, 60*, 84 - 90.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal Loss for Dense Object Detection. *Proceedings of the IEEE Int Conf Comput Vis*, *8*, 2980. http://arxiv.org/abs/1708.02002

Liu, Y., Palmedo, P., Ye, Q., Berger, B., & Peng, J. (2018). Enhancing Evolutionary Couplings with Deep Convolutional Neural Networks. *Cell systems*, *6*(1), 65–74.e3. https://doi.org/10.1016/j.cels.2017.11.014

Lunt, B., Szurmant, H., Procaccini, A., Hoch, J. A., Hwa, T., & Weigt, M. (2010). Inference of direct residue contacts in two-component signaling. *Methods in enzymology*, *471*, 17–41. https://doi.org/10.1016/S0076-6879(10)71002-8

Ma, W., Cao, Y., Bao, W., Yang, B., & Chen, Y. (2020). ACT-SVM: Prediction of Protein-Protein Interactions Based on Support Vector Basis Model. *Scientific Programming*, *2020*. https://doi.org/10.1155/2020/8866557

Maveyraud, L., & Mourey, L. (2020). Protein X-ray Crystallography and Drug Discovery. *Molecules (Basel, Switzerland)*, *25*(5), 1030. https://doi.org/10.3390/molecules25051030

Reuter, J. A., Spacek, D. V., & Snyder, M. P. (2015). High-throughput sequencing technologies. *Molecular cell*, *58*(4), 586–597. https://doi.org/10.1016/j.molcel.2015.05.004

Sehnal, D., Bittrich, S., Deshpande, M., Svobodová, R., Berka, K., Bazgier, V., Velankar, S., Burley, S. K., Koča, J., & Rose, A. S. (2021). Mol* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures. *Nucleic acids research*, *49*(W1), W431–W437. https://doi.org/10.1093/nar/gkab314

Skerker, J. M., Perchuk, B. S., Siryaporn, A., Lubin, E. A., Ashenberg, O., Goulian, M., & Laub, M. T. (2008). Rewiring the specificity of two-component signal transduction systems. *Cell*, *133*(6), 1043–1054. https://doi.org/10.1016/j.cell.2008.04.040

Vreven, T., Moal, I. H., Vangone, A., Pierce, B. G., Kastritis, P. L., Torchala, M., Chaleil, R., Jiménez-García, B., Bates, P. A., Fernandez-Recio, J., Bonvin, A. M., & Weng, Z. (2015). Updates to the Integrated Protein-Protein Interaction Benchmarks: Docking Benchmark Version 5 and Affinity Benchmark Version 2. *Journal of molecular biology*, *427*(19), 3031–3041. https://doi.org/10.1016/j.jmb.2015.07.016

Wang, S., Sun, S., Li, Z., Zhang, R., & Xu, J. (2017). Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. *PLoS computational biology*, *13*(1), e1005324. https://doi.org/10.1371/journal.pcbi.1005324

Weigt, M., White, R. A., Szurmant, H., Hoch, J. A., & Hwa, T. (2009). Identification of direct residue contacts in protein-protein interaction by message passing. *Proceedings of the National Academy of Sciences of the United States of America*, *106*(1), 67–72. https://doi.org/10.1073/pnas.0805923106

White, R. A., Szurmant, H., Hoch, J. A., & Hwa, T. (2007). Features of protein-protein interactions in two-component signaling deduced from genomic libraries. *Methods in enzymology*, *422*, 75–101. https://doi.org/10.1016/S0076-6879(06)22004-4

Wu, Q., Peng, Z., Anishchenko, I., Cong, Q., Baker, D., & Yang, J. (2020). Protein contact prediction using metagenome sequence data and residual neural networks. *Bioinformatics (Oxford, England)*, *36*(1), 41–48. https://doi.org/10.1093/bioinformatics/btz477

Xu J. (2019). Distance-based protein folding powered by deep learning. *Proceedings of the National Academy of Sciences of the United States of America*, *116*(34), 16856–16865. https://doi.org/10.1073/pnas.1821309116