Modelo neuronal basado en paso de mensajes para estimación de similaridad entre compuestos

Matias Gerard y Leandro Di Persia

Research Institute for Signals, Systems and Computational Intelligence (sinc(i)),
FICH–UNL/CONICET, Ciudad Universitaria UNL, (S3000) Santa Fe, Argentina.
mgerard@sinc.unl.edu.ar
www.sinc.unl.edu.ar

Resumen La búsqueda de vías metabólicas tiene como objetivo encontrar secuencias de reacciones que permitan transformar un sustrato dado en un producto de interés. Esta tarea puede abordarse como un problema de búsqueda en grafos, usando la estructura molecular de los compuestos y una medida de similaridad entre las estructuras para guiar la búsqueda. Sin embargo, los enfoques basados en esta idea resultan inútiles cuando se carece de la estructura, lo que impide calcular la similaridad. Por su parte, las redes neuronales en grafos han demostrado ser de gran utilidad como extractores de características en datos con estructuras no-euclideanas. Aquí presentamos un modelo neuronal basado en grafos, capaz de aprender representaciones de los compuestos a partir de características simples y de la topología de la red que los conecta. Estas características son luego empleadas para inferir la similaridad, sin que sea necesaria la estructura de los mismos en el proceso. Los resultados muestran que el modelo infiere correctamente la similaridad entre compuestos con estructura conocida, y genera estimaciones razonables para compuestos con estructura desconocida.

Keywords: Redes neuronales en grafos · Vías metabólicas · Similaridad entre compuestos.

1. Introducción

Las redes metabólicas son una parte fundamental de los sistemas biológicos, encargadas de transformar compuestos y generar energía. Están construidas por intrincadas relaciones entre compuestos, denominadas reacciones bioquímicas. Formalmente, las reacciones se describen mediante ecuaciones químicas típicas como $S(r) \leftrightarrow P(r)$, donde $S(r)$ y $P(r)$ corresponden a los sustratos y los productos, respectivamente [14].

Un desafío que enfrenta la bioinformática actualmente es la búsqueda y diseño de vías metabólicas. El objetivo es identificar la secuencia adecuada de reacciones necesarias para sintetizar nuevos compuestos a partir de otros dados. Típicamente esta tarea se ha abordado mediante métodos basados en grafos. Así, el primer paso es modelar los compuestos y las reacciones como un grafo apropiado. Una representación empleada habitualmente es el grafo de compuestos $\mathcal{G} = \{v, e\}$, donde los nodos v representan compuestos y los arcos e conectan

sustratos $S(r)$ y productos $P(r)$ de una misma reacción r [2]. El siguiente paso es buscar un camino sobre el grafo que conecte la fuente con el compuesto objetivo utilizando algún método de búsqueda [7,13,8]. El problema principal consiste en evitar los compuestos denominados *pool*, como es el caso del agua y el ATP (almacenamiento de energía), implicados en un gran número de reacciones. Debido a su gran conectividad en el grafo, estos son incluidos frecuentemente como intermediarios en las soluciones y producen vías biológicamente inviables.

Entre las estrategias propuestas para abordar el manejo de los compuestos *pool* se encuentra el uso de la estructura molecular de los compuestos para guiar la búsqueda [15,16]. Este enfoque busca maximizar la similaridad entre compuestos consecutivos durante la construcción del camino y con el compuesto final a producir. La principal ventaja es que no se requiere la identificación y filtrado de los compuestos *pool*, y que en algunos casos podría ser de interés definirlos como inicio o fin de una búsqueda. Desafortunadamente, existen compuestos para los que se desconoce la estructura molecular, lo que dificulta el uso de esta estrategia. Por ejemplo, los compuestos `C00138`[1] y `C00139`[2] de la base de datos KEGG[12] corresponde a la proteína Ferredoxina en estado reducido y oxidado, respectivamente. Aunque se trata del mismo compuesto con diferencias mínimas, no es posible el cálculo de similaridad mediante los métodos habituales de comparación de huellas digitales [3]. Esto se debe a que su funcionamiento se basa en la comparación de características que extraen a partir de la estructura de los compuestos.

En la última década, el aprendizaje profundo se ha convertido en un elemento básico en la caja de herramientas de aprendizaje automático de muchas áreas de investigación [1,19,4], como es el caso del análisis de redes biológicas [10,11,5]. Entre los modelos más recientes se encuentran las redes neuronales en grafos [21,18], cuya principal característica es la habilidad para manipular y procesar la información con estructuras no euclideanas. En particular, buscan extender las características de las redes convolucionales tradicionales a este tipo de datos [20]. Recientemente, Gilmer y col. [9] propusieron un nuevo paradigma dentro de las redes neuronales en grafos, que fue empleado exitosamente para predecir propiedades químicas de compuestos. Este marco conceptual, denominado red neuronal de paso de mensajes (MPNN, del inglés), puede verse como una generalización de las redes convolucionales en grafos. Su funcionamiento es similar, ya que se basa en la actualización de las características de un nodo dado del grafo en base a la combinación de las características de sus vecinos y de las propiedades de los arcos que los conectan. MPNN involucra tres pasos comunes: (i) construcción del mensaje, donde las características de cada nodo o arco se propagan a los vecinos de acuerdo a la estructura del grafo a través de un vector de mensajes; (ii) agregación, donde los vectores de mensajes para cada nodo se combinan en una única representación; (iii) actualización, donde las características de cada nodo son actualizadas en base a las características propias y al vector obtenido en la etapa de agregación. Las características resultantes de la etapa

[1] `https://www.genome.jp/entry/C00138`
[2] `https://www.genome.jp/entry/C00139`

(iii) pueden luego ser utilizadas por una capa densa para inferir una propiedad especificada. Al igual que las redes convolucionales en grafos, la MPNN es capaz de actuar como un extractor de características útiles de nodos y arcos a partir de la estructura del grafo, evitando la ingeniería de características frecuentemente encontrada en la etapa de preprocesamiento.

Claramente, el modelado de las redes metabólicas como grafos de compuestos vuelve explícitas las relaciones que existen entre los mismos. A su vez, incorporando información que caracterice unívocamente a cada nodo podría ser posible aplicar un modelo basado en MPNN para inferir características que resulten de utilidad para inferir la similaridad entre compuestos. En base a estas ideas, en este trabajo se propone un nuevo modelo neuronal basado en el paradigma del paso de mensajes para inferir la similaridad entre compuestos.

La organización del trabajo es la siguiente. La Sección 2 presenta el modelo y sus componentes. En la Sección 3 se describen los datos empleados y cómo se construye el dataset. En la Sección 4 se presentan los resultados donde se analiza el desempeño del modelo frente a diferentes modificaciones en sus partes. También se incluye un breve análisis de la inferencia de la similaridad entre compuestos con estructuras desconocidas. Finalmente, en la Sección 5 se presentan las conclusiones del trabajo.

2. Descripción del modelo

2.1. Grafos de compuestos y subgrafos para estimación de similaridad

Sea $\mathcal{G} = \{v, e\}$ el grafo de compuestos que describe a una red metabólica, donde v es el conjunto de nodos que representan a los N compuestos que participan de la red, y e son los arcos que conectan sustratos y productos de una misma reacción. Sea también $\mathbf{X} \in \mathcal{R}^{NxD}$ la matriz de características asociada a los compuestos de $\mathcal{G}$, donde D es el número de características que describen a cada nodo. Se define $\mathcal{G}'_{i,j} = \{v', e'\} \subset \mathcal{G}$, como el subgrafo inducido por los nodos i y j, siendo $v' = \{i, j, \mathcal{N}(i), \mathcal{N}(j)\}$ los nodos que lo componen, e' las conexiones que existen entre ellos, y $\mathcal{N}(k)$ el conjunto de nodos denominados *vecinos directos* que comparten una arista con el nodo k. Además, se define $\mathbf{X}' \in \mathcal{R}^{nxD}$ como la matriz de características asociadas a los $n = |v'|$ nodos que componen el subgrafo $\mathcal{G}'_{i,j}$.

2.2. Red neuronal de paso de mensajes

La red neuronal de paso de mensajes (MPNN, del inglés) es una arquitectura de aprendizaje profundo diseñada para su implementación en contextos químicos, farmacéuticos y de ciencia de los materiales [9]. Puede verse como una generalización de otros modelos, dado que propone un mecanismo general para el intercambio de información entre nodos de una red. De forma resumida, cada capa de una MPNN gestiona el intercambio de información de cada nodo con

sus vecinos mediante la construcción de vectores de mensajes, que luego son utilizados para actualizar la información de cada nodo. Además, la repetición de este proceso mediante la concatenación de múltiples capas o la realimentación de la salida permite que los nodos se actualicen con información de vecinos cada vez más lejanos. Dado que la información es transformada en cada paso mediante redes neuronales, el modelo resultante puede entrenarse para optimizar las representaciones para cada tarea considerada.

Formalmente, las MPNN realizan tres operaciones principales: paso de mensajes, actualización de nodos y lectura. El uso de una red neuronal de paso de mensajes implica la actualización iterativa del conjunto de características $x_v \in \mathcal{R}^D$ de cada nodo v. El paso de mensajes y la actualización de nodos se hacen de acuerdo a las siguientes ecuaciones:

$$m_v^{(t+1)} = \sum_{w \in \mathcal{N}(v)} M_t \left(x_v^{(t)}, x_w^{(t)}, e_{vw} \right) \tag{1}$$

$$x_v^{(t+1)} = U_t \left(x_v^{(t)}, m_v^{(t+1)} \right) \tag{2}$$

donde M_t es la función de mensajes, U_t es la función de actualización del nodo, $\mathcal{N}(v)$ es el conjunto de vecinos directos del nodo v en el grafo $\mathcal{G}$, e_{vw} es el arco que conecta los nodos v y w, $x_v^{(t)}$ es el conjunto de características del nodo v en el tiempo t, y $m_v^{(t+1)}$ es un vector de mensajes correspondiente. Nótese que M_t y U_t pueden variar con el paso de t. Para cada nodo v, los mensajes pasarán desde sus vecinos y se agregarán como el vector de mensajes $m_v^{(t+1)}$ de su entorno. A continuación, el estado oculto x_v se actualiza mediante el vector de mensajes. La fórmula de la función de lectura se muestra en la ecuación 3:

$$\widetilde{y} = R \left(\{ x_v^{(K)} | v \in \mathcal{G} \} \right) \tag{3}$$

donde $\widetilde{y}$ es un vector de características de longitud fija que resume la información de los nodos de $\mathcal{G}$, y R es una función de lectura invariante al ordenamiento de los nodos, una característica importante que permite que la MPNN sea invariante al isomorfismo del grafo. El vector de características $\widetilde{y}$ del grafo puede pasarse a una capa totalmente conectada para obtener luego una predicción. Todas las funciones M_t, U_t y R son redes neuronales y sus pesos se aprenden durante el entrenamiento.

2.3. Modelo neuronal propuesto

El modelo propuesto emplea una MPNN como extractor de características, que luego son procesadas mediante capas completamente conectadas para realizar la inferencia. En particular, la MPNN utilizada consiste en una única capa y sólo emplea las Ec. 1 y 2 para la actualización de las características. Mientras que la tarea descripta en la Ec. 1 es realizada combinando la aplicación de un perceptrón multicapa (MLP, en inglés) y un mecanismo basado en atención, la Ec. 2 consiste simplemente en la concatenación de las características obtenidas

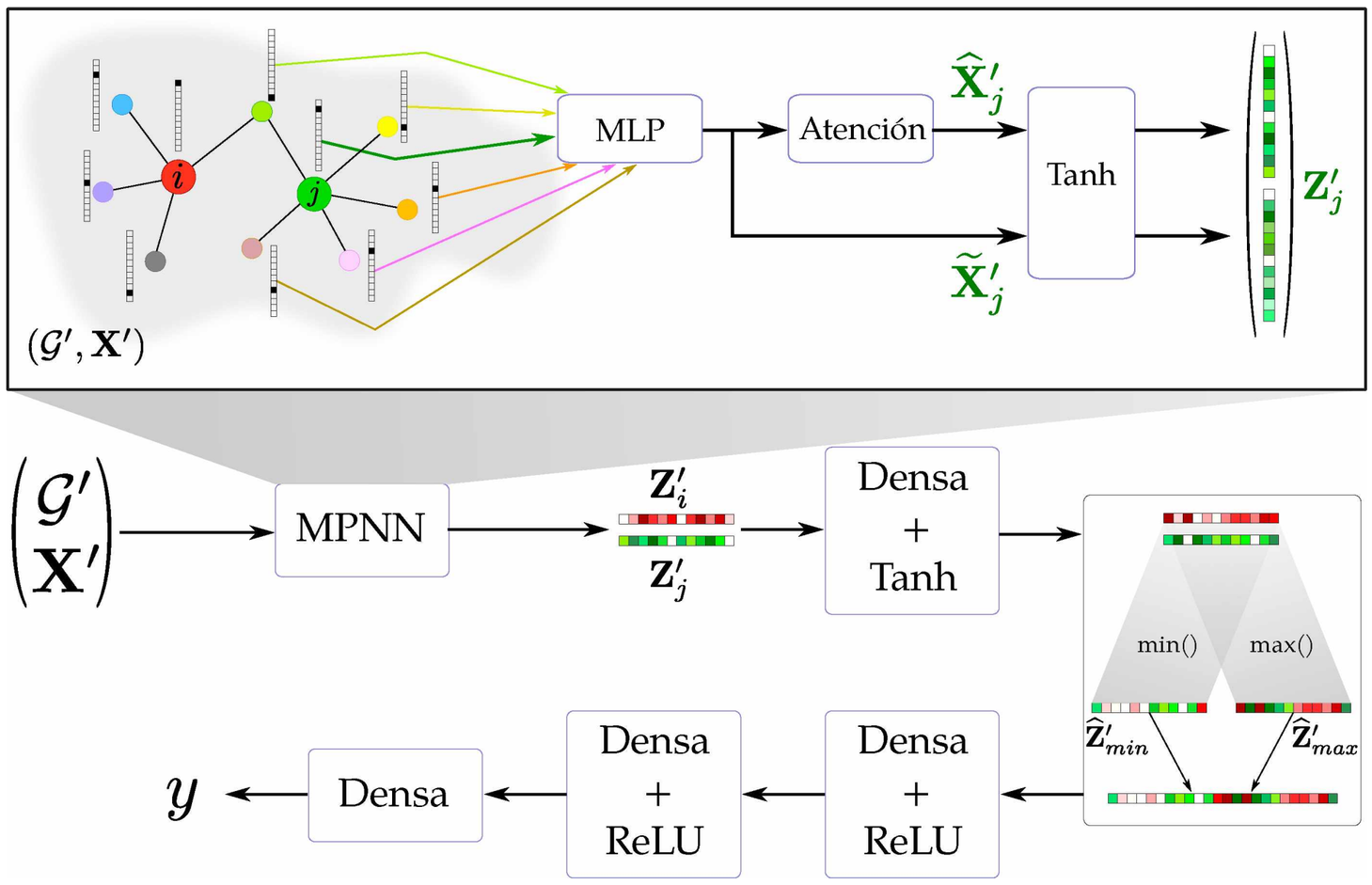

Figura 1. Arquitectura neuronal propuesta.

de cada una de las salidas de las redes empleadas previamente y su transformación mediante una función no lineal. La Figura 1 presenta un esquema del modelo propuesto.

Extracción de características. Este modelo recibe como entrada a la MPNN un subgrafo $\mathcal{G}'$ como se describió en la Sección 2.1, y las características $\mathbf{X}' \in \mathcal{R}^{nxD}$ asociadas a sus nodos. En particular, en este modelo se emplea una codificación simple de tipo one-hot para describir las características de cada compuesto. Esta capa realiza dos etapas de procesamiento. Primero, las características $\mathbf{X}'$ son proyectadas por un MLP a un espacio más adecuado para la inferencia, obteniendo una nueva representación $\widetilde{\mathbf{X}}'$. Este paso permite manipular el grado de compresión de la información en la salida mediante la generación de representaciones más compactas, permitiendo además convertir representaciones ralas a un espacio denso. Luego, la información de cada nodo $j \in \mathcal{G}'$ es actualizada mediante el mecanismo de atención descripto en el Algoritmo 1, que será explicado a continuación.

Para un nodo j dado, el Algoritmo 1 proyecta inicialmente las características $\widetilde{\mathbf{X}}'_j$ de este nodo y las características $\widetilde{\mathbf{X}}'_{m \in \mathcal{N}(j)}$ de su vecindad $\mathcal{N}(j)$ a tres nuevos espacios de dimensión idéntica a la original. Esta proyección no lineal se realiza empleando las matrices $\mathbf{W}^Q$, $\mathbf{W}^K$ y $\mathbf{W}^V$, y una función sigmoidea σ. Nótese que mientras $\mathbf{W}^K$ y $\mathbf{W}^V$ son empleadas para proyectar las características $\widetilde{\mathbf{X}}' = \left[\widetilde{\mathbf{X}}'_j, \widetilde{\mathbf{X}}'_{m \in \mathcal{N}(j)}\right]$ de todos los nodos del subgrafo $\mathcal{G}'$, sólo $\widetilde{\mathbf{X}}'_j$ es proyectado

Algoritmo 1: Actualización de características mediante atención.

Input: $\widetilde{\mathbf{X}}'_j$, $\widetilde{\mathbf{X}}'_{m\in N(j)}$.
Output: $\widehat{\mathbf{X}}'_j$.
begin
  $\widetilde{\mathbf{X}}' = \left[\widetilde{\mathbf{X}}'_j, \widetilde{\mathbf{X}}'_{m\in\mathcal{N}(j)}\right]$ // Unión de características
  $\mathbf{q} = \sigma\left(\widetilde{\mathbf{X}}'_j \cdot \mathbf{W}^Q\right)$ // Características del nodo central // proyectadas al espacio Q
  $\mathbf{K} = \sigma\left(\widetilde{\mathbf{X}}' \cdot \mathbf{W}^K\right)$ // Características proyectadas al espacio K
  $\mathbf{V} = \sigma\left(\widetilde{\mathbf{X}}' \cdot \mathbf{W}^V\right)$ // Características proyectadas al espacio V
  $\mathbf{w} = softmax\left(\mathbf{q} \cdot \mathbf{K}^T\right)$ // Cálculo de similaridad normalizada
  $\widehat{\mathbf{X}}'_j = \sum_{m\in\{j\cup\mathcal{N}(j)\}} \mathbf{w}_m \cdot \mathbf{V}_m$ // Combinación ponderada

al espacio definido por $\mathbf{W}^Q$. A continuación, se calcula la similaridad entre la proyección $\mathbf{q}$ de las características del nodo j y las proyecciones $\mathbf{K}$ de las de su vecindad, y se normaliza este vector de pesos empleando una función *softmax*. El vector $\mathbf{w}$ es luego empleado para construir la nueva representación $\widehat{\mathbf{X}}'_j$ del nodo central, mediante la combinación ponderada de las representaciones $\mathbf{V}$ de los nodos de la vecindad. Una vez actualizada la información, la MPNN retorna un nuevo vector de características $\mathbf{Z}'_j = Tanh\left(\left[\widehat{\mathbf{X}}'_j, \widetilde{\mathbf{X}}'_j\right]\right)$ para cada nodo j, que se obtiene luego de aplicar una función no-lineal al vector de características construido mediante la concatenación de los vectores de características procedentes de la salida del MLP y del proceso de atención.

Procesamiento de características e inferencia. Las representaciones aprendidas por la MPNN son luego empleadas para inferir la similaridad entre los compuestos de los nodos i y j usados para construir $\mathcal{G}'$. Para esto sólo se emplean las representaciones $\mathbf{Z}'_i$ y $\mathbf{Z}'_j$ de estos nodos. Estas representaciones son nuevamente procesadas por una capa completamente conectada y proyectadas de forma no lineal, y sin modificación de la dimensión de salida, a un nuevo espacio. A continuación, se calcula el valor mínimo y máximo para cada característica considerando ambas representaciones, obteniendo dos nuevos conjuntos de características $\widehat{\mathbf{Z}}'_{min}$ y $\widehat{\mathbf{Z}}'_{max}$. Estos son luego concatenados para generar una representación única $\widehat{\mathbf{Z}}'$ compuesta por el doble características para el par de nodos considerado. Este paso permite que las etapas posteriores de la inferencia sean invariantes al orden en que se presentan ambos nodos.

La salida concatenada $\widehat{\mathbf{Z}}'$ es procesada por dos capas completamente conectadas, cada una generando una salida con dimensión igual a la mitad de la entrada, seguida de una modificación no lineal de los valores en cada caso. La última capa, completamente conectada, genera una única salida lineal y con el valor de la inferencia.

3. Datos empleados y construcción de patrones

Se emplearon las 30 reacciones que componen la vía metabólica de la glicólisis[3] en la bacteria *Escherichia coli* (*ecoli*). Los datos fueron extraídos de la base de datos de KEGG (v95.2) [12]. Empleando estas reacciones se construyó el grafo $\mathcal{G}$, conteniendo un total de 46 nodos y 126 conexiones entre ellos. También se descargó para cada compuesto la estructura molecular, cuando estaba disponible, de la base de datos PubChem[4]. El manejo y procesamiento de las estructuras se realizó con la librería RDKit[5]. La similaridad entre pares de compuestos se calculó empleando el coeficiente de Tanimoto [3], definido como:

$$T(\mathbf{m}_i, \mathbf{m}_j) = \frac{\sum_k (\mathbf{m}_i^k \wedge \mathbf{m}_j^k)}{\sum_k (\mathbf{m}_i^k \vee \mathbf{m}_j^k)} \tag{4}$$

donde $\wedge$ y $\vee$ son los operadores binarios *and* y *or*, respectivamente, y $\mathbf{m}_i$ y $\mathbf{m}_j$ son representaciones binarias de las estructuras de los compuestos i y j. Cada una de estas representaciones se construye evaluando la presencia de una amplia variedad de características en las estructuras de los compuestos (ej. número de átomos de hidrógeno, número de anillos). El coeficiente de Tanimoto toma valores en el rango $[0, 1]$ y calcula la proporción de características compartidas entre ambas estructuras. En este trabajo, las representaciones fueron calculadas empleando el algoritmo de Morgan [17,6] provisto por la librería RDKit[6]. En todos los casos se empleó un radio $r = 3$ y 2048 bits para cada estructura.

El dataset de entrenamiento se construyó considerando todos los pares de compuestos (i, j) para los que se disponía de la estructura de ambos. En cada caso, el subgrafo $\mathcal{G}'$ asociado se construyó considerando los nodos i, j y los vecinos respectivos $N(i)$ y $N(j)$. También se consideraron los casos donde $i = j$, independientemente de que la estructura se encontrara disponible. El conjunto de características iniciales $\mathbf{X}$ empleadas para describir a los compuestos se construyó como una matriz identidad, considerando una codificación one-hot para cada compuesto.

4. Resultados y discusión

Los experimentos realizados en esta Sección se llevaron a cabo empleando 20000 épocas de entrenamiento. Se estableció un criterio de detención temprana en 5000 épocas sin mejora en los valores de validación. Este límite elevado se definió en base a experimentos preliminares, donde se observó que el error de validación frecuentemente experimenta un incremento importante antes de descender nuevamente. En cada partición se utilizó el 30 % de los datos de entrenamiento para validación. Los experimentos se llevaron a cabo en una PC

[3] `https://www.genome.jp/pathway/eco00010`
[4] `https://pubchem.ncbi.nlm.nih.gov/`
[5] `https://www.rdkit.org`
[6] `https://www.rdkit.org/docs/GettingStartedInPython.html#morgan-fingerprints-circular-fingerprints`

i7-7700 con 64 GB de memoria RAM. El modelo fue implementado utilizando pytorch 1.8.1, pytorch geometric 1.7.0 y pytorch-lightning 1.2.10. El entrenamiento se realizó empleando el optimizador Adam, con una tasa de aprendizaje de $5 \cdot 10^{-4}$ y un tamaño de mini-batch de 16 patrones. El entrenamiento de todos los modelos se realizó empleando el error cuadrático como función de costo.

4.1. Evaluación del tamaño y tipo de red usado en la densificación de la MPNN

Uno de los principales componentes del modelo es el perceptrón multicapa que forma parte de la MPNN, ya que permite manipular el grado de compresión de la información asociada a cada compuesto. Esto puede resultar particularmente útil cuando crece el número de compuestos considerados, ya que el incremento en el tamaño de la representación conlleva un aumento en el número de pesos entrenables en el modelo propuesto. Además, cuando los vectores de características asociados a los compuestos son ralos, este componente también es responsable de la transformación a una representación densa.

Para evaluar el efecto de este componente sobre el desempeño del modelo se consideraron tres arquitecturas: (i) una única capa densa, sin función de activación; (ii) una única capa densa aplicando la función no lineal *Tanh* a la salida; (iii) el modelo $\widetilde{\mathbf{X}}'_j = \mathbf{W}_2 \cdot \sigma(\mathbf{W}_1 \cdot \mathbf{X}'_j)$, compuesto de dos capas y una función no lineal *Tanh* a la salida de la primera capa. Además, este modelo realiza una compresión de 50 % a la salida de la primera capa ($\mathbf{W}_1$), independientemente del tamaño de la salida final generada ($\mathbf{W}_2$). A su vez, para cada propuesta se generaron salidas finales con diferente grado de compresión. Así, una compresión de 50 % implica que el MLP devolverá un vector de 23 elementos cuando reciba como entrada un vector con 46 características. En todos los experimentos se empleó validación cruzada con 5 particiones, manteniendo fija la arquitectura de la MPNN (como se describe en la Figura 1) y realizando sólo modificaciones en la estructura del MLP. Para cada una de las 12 configuraciones posibles se llevó a cabo el entrenamiento del modelo completo. La Figura 2 presenta los resultados para las dos medidas de desempeño consideradas, obtenidas en cada una de las 5 particiones de test evaluadas con cada configuración experimental. La Figura 2a presenta los resultados para la suma de los errores absolutos acumulados (SEA), y la Figura 2b la media de los errores absolutos (MEA). Nótese que mientras que el SEA mide el error absoluto acumulado en las particiones de test, el MEA brinda una idea del error promedio cometido por el modelo al momento de inferir un valor de similaridad entre pares de compuestos.

Del análisis de la Figura 2 se observa que el incremento en el número de capas no se traduce en una reducción del SEA o el MEA. Además, en términos generales, mientras que la compresión en la salida genera representaciones más compactas, ésta conduce a un aumento del error en la predicción de la similaridad. Claramente, el SEA y MEA tienden a aumentar al aplicar un factor de compresión mayor a 25 % en los 3 modelos de MLP estudiados. Por otra parte, se observa que la incorporación de la función no lineal tiende a generar mayor

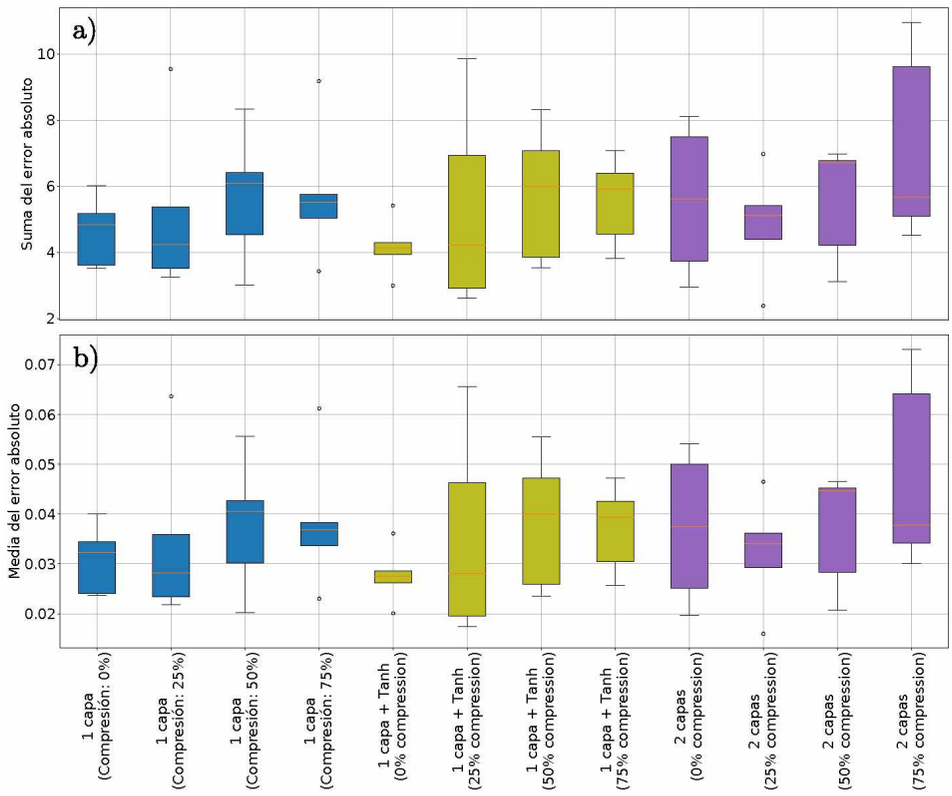

Figura 2. Medidas de desempeño evaluadas en la partición de test para distintos esquemas de densificación. a) Suma del error absoluto. b) Promedio del error absoluto. La línea roja en el cuerpo de las cajas indica el valor de la mediana.

dispersión en los resultados obtenidos en las distintas particiones, y sólo mejora el desempeño del modelo que emplea una única capa densa sin compresión.

Para analizar en mayor detalle el funcionamiento del modelo que emplea una capa densa y función no lineal, se graficaron los valores de similaridad real (calculados con la Ec. 4)y predicha para todos los compuestos en uno de los experimentos llevados a cabo. Para esto, se tomó el modelo guardado durante el entrenamiento que produjo el menor error de validación, y se determinó la similaridad para todos los pares de compuestos. La Figura 3 presenta los resultados para las particiones de entrenamiento, validación y test empleadas. La línea diagonal indica el caso ideal donde los valores reales y predichos son iguales.

Puede observarse que el modelo analizado predice con gran exactitud y precisión la similaridad entre la mayor parte de los compuestos en las partición de entrenamiento. Sin embargo, se observa una tendencia a subestimar la similaridad en el rango $[0{,}3; 1]$, que podría deberse al reducido número de casos de los que dispone para aprender a realizar correctamente la inferencia. También puede observarse un buen desempeño en la partición de validación, presentando nue-

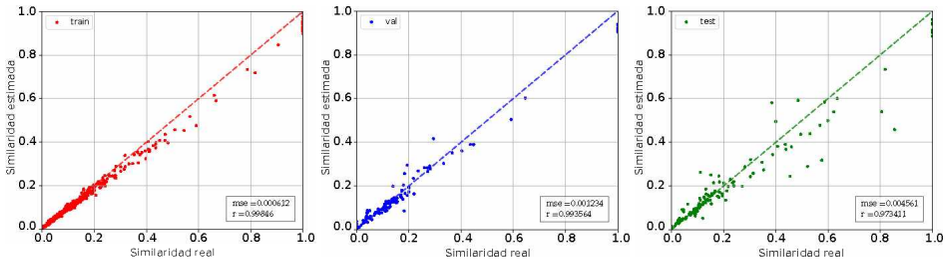

Figura 3. Comparación de los valores de similaridad calculados con la Ec. 4 e inferidos para las particiones de entrenamiento, validación y test. La diagonal representa la situación donde los valores calculados e inferidos son iguales. *mse*: error cuadrático medio; r: coeficiente de correlación de Pearson.

vamente un bajo grado de dispersión respecto al caso ideal. Sin embargo, en la partición de test resulta más evidente este problema, ya que la dispersión de los valores aumenta en el mismo rango considerado. Es posible que la incorporación de algún esquema de ponderación de los errores en base al número de patrones disponibles de acuerdo a la distribución de valores de similaridad contribuya a reducir esta dispersión. Un aspecto a resaltar es que el modelo presenta dificultades para inferir la similaridad de un compuestos consigo mismo. Esto se observa en los 3 gráficos, que presentan patrones cuya similaridad calculada con la Ec. 4 es 1,0 pero que el modelo subestima. Sin embargo, es posible que el uso de un esquema de ponderación diferencial de los errores pueda contribuir a reducir este efecto.

Como puede verse, el uso de una única capa densa y la aplicación de una función no lineal produce las mejores estimaciones de similaridad. Sin embargo, la dispersión encontrada en las estimaciones de similaridad para valores mayores a 0,3 indica que debe implementarse alguna estrategia para mejorar el desempeño del algoritmo en esa zona.

4.2. Predicción de similaridad en compuestos de estructura desconocida

Para evaluar la calidad de los resultados de similaridad inferidos para compuestos con estructura desconocida, se entrenó el modelo utilizando 80 % de los patrones para entrenamiento y reservando 20 % para validación. Luego se seleccionaron dos reacciones del dataset que contienen compuestos con estructura desconocida y se realizó un análisis cualitativo de la calidad de las predicciones realizadas. La reacción `R01196` involucra los compuestos `C00138` y `C00139` sin estructura conocida. Además, la estructura del compuesto `C00024` está constituída principalmente por `Coa` (`C00010`). En cambio, la reacción `R03270` contiene los compuestos `C15972` y `C16255`, los cuales son dos representantes de una amplia familia. Si bien se conoce parcialmente la estructura, ésta varía según la cadena lateral -R de cada compuesto. Esto hace que la información de la estructura

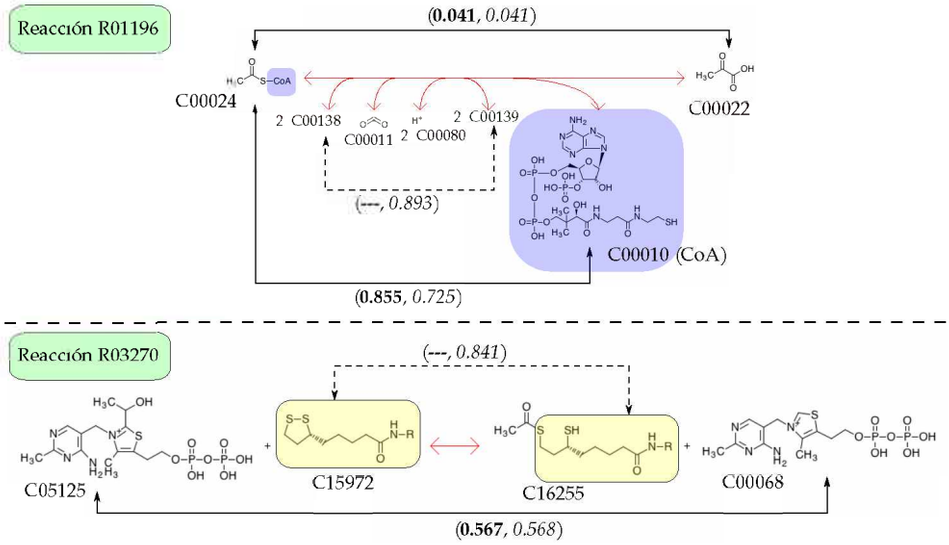

Figura 4. Reacciones del dataset que involucran compuestos con estructura molecular desconocida. Las flechas indican entre qué pares de compuestos se está estimando la similaridad. Sobre cada flecha se indica el valor de similaridad calculada con la Ec. 4 (negrita) y el valor inferido (cursiva). En línea de trazos se indican compuestos con estructura desconocida entre los que se infiere la similaridad. Las cajas azul y amarilla indican regiones compartidas entre compuestos.

no pueda ser procesada por los métodos de huellas digitales para calcular la similaridad. La Figura 4 presenta un esquema de las reacciones analizadas.

Como puede observarse, la reacción `R01196` descompone el sustrato `C00024` en los productos `C00022` y `C00010`. Como resulta esperable, `C00024` presenta una elevada similaridad (real: 0,855, inferida: 0,725) con el producto `C00010` dado que su estructura representa la mayor parte del sustrato. Si bien sería deseable una predicción más precisa, el valor predicho igualmente indica que ambos compuestos comparten una proporción importante de la estructura. Así mismo, la similaridad entre `C00024` y `C00022` es baja (real: 0,041, inferida: 0,041) debido a la poca contribución de la estructura del producto en el sustrato. Por otra parte, la similaridad inferida para los compuestos `C00138` y `C00139` es 0,893, lo que indica que comparten una proporción importante de sus estructuras. Este resultado es acorde a lo que se conoce para estos compuestos, que sólo se diferencian en su estado de oxidación. En cuanto a la reacción `R03270`, los valores de similaridad calculados con la Ec. 4 (0,567) e inferida (0,568) para los compuestos `C05125` y `C0068` presentan diferencias mínimas en el tercer decimal, como es de esperarse debido al uso de la mayor parte del dataset en el entrenamiento. Por otro lado, la similaridad inferida entre los compuestos `C15972` y `C16255` es 0,841, indicando una elevada similaridad entre sus estructuras. Tal y como puede observarse, esto resulta esperable dado que sólo difieren en un pequeño grupo de átomos que adquiere `C15972` luego de la reacción.

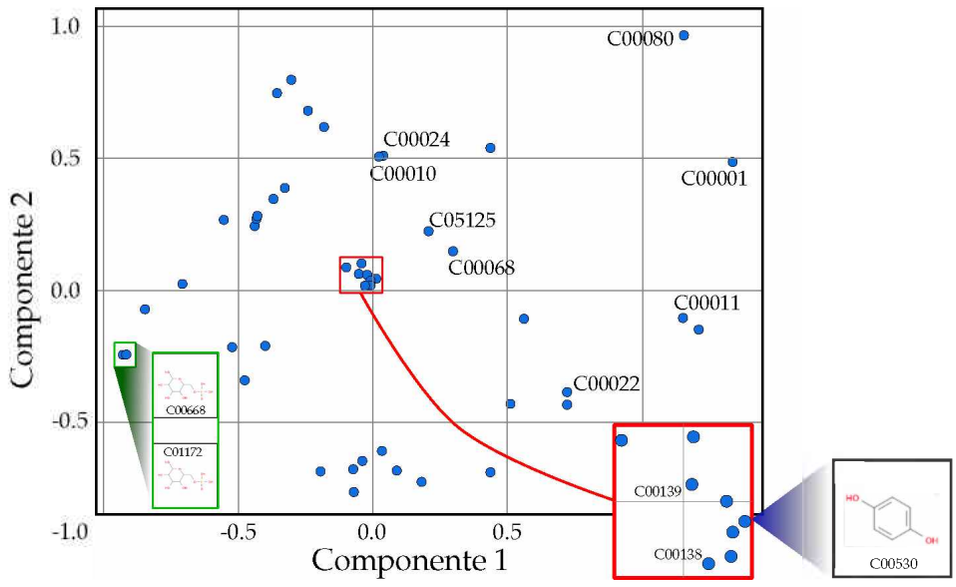

Figura 5. Proyección en 2 dimensiones, mediante PCA, de los vectores de características de los nodos del grafo de compuestos. Se incluyen las proyecciones para compuestos con estructura conocida y desconocida.

Otro aspecto que se analizó fue la estructura de las representaciones internas generadas por el modelo para describir los compuestos. Para eso, se tomaron los vectores de características de los nodos luego de ser procesados por el MLP dentro de la MPNN. Dado que estas transformaciones son optimizadas durante el entrenamiento y guiadas por la similaridad entre compuestos, resulta esperable que las mismas capturen información geométrica que describa la relación entre los mismos. Con el objetivo de visualizar estas relaciones se construyó la Figura 5, que presenta una proyección en 2 dimensiones usando PCA de los vectores de características de todos los compuestos que forman parte del grafo.

Un aspecto a destacar es la formación de agrupamientos de compuestos, los cuales se caracterizan por contener compuestos entre los que existe una elevada similaridad. Así, los pares de compuestos `C00010` – `C00024`, `C00138` – `C00139` y `C00068` – `C05125` se encuentran cerca en este espacio, tal y como es de esperar dado que presentan una importante similitud (ver Figura 4). También esto ocurre para los compuestos `C00668` – `C01172`, que son sustrato y producto de la reacción `R02739` que no modifica la composición de los compuestos (similaridad calculada con la Ec. 4: 1,00, similaridad inferida: 1,0). En cambio, `C00022` y `C00024` se encuentran alejados dado que presentan una baja similitud a causa de que comparten una muy pequeña porción de sus estructuras. Esto también se observa para los compuestos `C00011` (dióxido de carbono) y `C00001` (agua), que no comparte similaridad en sus estructuras (real: 0,0, predicha: 0,0). Otro hecho interesante es que todos los compuestos con estructura desconocida se en-

cuentran agrupados en el recuadro rojo, a excepción del compuesto C00530. Esto resultados muestran que el modelo es capaz de generar representaciones espaciales de los compuestos que capturan adecuadamente las relaciones de similaridad que existen entre ellos.

5. Conclusión

En este trabajo se presenta un nuevo modelo basado en redes neuronales de paso de mensajes para inferir la similaridad entre compuestos metabólicos. Empleando representaciones one-hot para describirlos, este modelo es capaz de aprender automáticamente nuevas representaciones más adecuadas para realizar la inferencia de similaridad. Para esto utiliza un mecanismo de atención que contempla la topología de la red metabólica en la que participan los compuestos. Los resultados muestran que los valores de similaridad inferidos difieren muy poco de los valores estimados con métodos que utilizan la estructura. En base a resultados cualitativos obtenidos para algunas reacciones analizadas, se observa que los valores de similaridad obtenidos para compuestos con estructura desconocida se encuentran acordes a lo esperado. Adicionalmente se encontró que las representaciones generadas por el modelo neuronal para todos los compuestos presenta una estructura geométrica que logra capturar las relaciones de similaridad. El siguiente paso será extender el análisis a un mayor número de vías metabólicas, contemplando incluso casos de vías metabólicas de organismos completos. También se evaluarán mecanismos para proporcionar un valor de incerteza asociado a las predicciones que se realicen.

Agradecimientos

Este trabajo fue financiado por ANPCyT (PICT 2019-03420) y UNL (CAI+D 2020 #50620190100115LI).

Referencias

1. Abu-El-Haija, S., Perozzi, B., Al-Rfou, R., Alemi, A.A.: Watch your step: Learning node embeddings via graph attention. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31, pp. 9180–9190. Curran Associates, Inc. (2018)
2. Arita, M.: From metabolic reactions to networks and pathways. Methods Mol. Biol. **804**, 93–106 (2012)
3. Bajusz, D., Rácz, A., Héberger, K.: Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? Journal of Cheminformatics **7**(1), 20 (2015). https://doi.org/10.1186/s13321-015-0069-3
4. Bugnon, L., Yones, C., Raad, J., Gerard, M., Rubiolo, M., Merino, G., Pividori, M., Di Persia, L., Milone, D., Stegmayer, G.: DL4papers: a deep learning approach for the automatic interpretation of scientific articles. Bioinformatics p. btaa111 (2020). https://doi.org/10.1093/bioinformatics/btaa111

5. Cang, Z., Wei, G.W.: TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. PLoS Comput Biol **13**(7), e1005690 (2017). https://doi.org/10.1371/journal.pcbi.1005690
6. Capecchi, A., Probst, D., Reymond, J.: One molecular fngerprint to rule them all: drugs, biomolecules, and the metabolome. J. Cheminform **12**(1), 43 (2020). https://doi.org/10.1186/s13321-020-00445-4
7. Gerard, M.F., Stegmayer, G., Milone, D.H.: Evolutionary algorithm for metabolic pathways synthesis. Biosystems. **144**, 55–67 (Jun 2016)
8. Gerard, M., Stegmayer, G., Milone, D.: Metabolic pathways synthesis based on ant colony optimization. Sci. Rep. **8**, 16398 (2018). https://doi.org/10.1038/s41598-018-34454-z
9. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 1263–1272. PMLR (06–11 Aug 2017)
10. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: A survey. Knowledge-Based Systems **151**, 78–94 (2018). https://doi.org/10.1016/j.knosys.2018.03.022
11. Grover, A., Leskovec, J.: node2vec: Scalable Feature Learning for Networks. In: KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery (ACM) (2016). https://doi.org/10.1145/2939672.2939754
12. Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., Morishima, K.: KEGG: new perspectives on genomes, pathways, diseases and drugs. Nucleic Acids Res. **45**(D1), D353–D361 (4 Jan 2017)
13. Kim, S.M., Peña, M.I., Moll, M., Bennett, G.N., Kavraki, L.E.: A review of parameters and heuristics for guiding metabolic pathfnding. J Cheminform **9**(1), 51 (2017). https://doi.org/10.1186/s13321-017-0239-6
14. Lacroix, V., Fernandes, C.G., Sagot, M.F.: Motif search in graphs: application to metabolic networks. IEEE/ACM Trans. Comput. Biol. Bioinform. **3**(4), 360–368 (Oct 2006)
15. McShan, D.C., Rao, S., Shah, I.: PathMiner: predicting metabolic pathways by heuristic search. Bioinformatics **19**(13), 1692–1698 (1 Sep 2003)
16. Rahman, S.A., Advani, P., Schunk, R., Schrader, R., Schomburg, D.: Metabolic pathway analysis web service (pathway hunter tool at CUBIC). Bioinformatics **21**(7), 1189–1193 (2005)
17. Rogers, D., Hahn, M.: Extended-Connectivity Fingerprints. J. Chem. Inf. and Model. **50**(5), 742–754 (2010). https://doi.org/10.1021/ci100050t
18. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The Graph Neural Network Model. IEEE Transactions on Neural Networks **20**(1), 69–80 (2009). https://doi.org/10.1109/TNN.2008.2005605
19. Stegmayer, G., Di Persia, L., Rubiolo, M., Gerard, M., Pividori, M., Yones, C., Bugnon, L., Rodriguez, T., Raad, J., Milone, D.: Predicting novel microRNA: a comprehensive comparison of machine learning approaches. Briefings in Bioinformatics **20**(5), 1607–1620 (2019). https://doi.org/10.1093/bib/bby037
20. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems pp. 1–21 (2020). https://doi.org/10.1109/TNNLS.2020.2978386
21. Zhang, S., Tong, H., Xu, J., Maciejewski, R.: Graph convolutional networks: a comprehensive review. Computational Social Networks **6**(11), 1–23 (2019). https://doi.org/10.1186/s40649-019-0069-y