Neural model-based similarity prediction for compounds with unknown structures

Eugenio Borzone, Leandro Ezequiel Di Persia, Matias Gerard

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

Abstract. Compounds similarity analysis is widely used in many areas related to cheminformatics. Its calculation is straightforward when compounds structures are known. However, there are no methods to get similarity when this information is not available. Here we propose a novel approach to solve this problem. It generates compound representations from metabolic networks, and are use a neural network to predict similarity. The results show that generated embeddings preserve the neighborhood of the original metabolic graph, i.e. compounds participating into the same reactions are close together in the embedding space. Results for compounds with known structures show that the proposal allows to estimate the similarity with an error of less than 10%. In addition, a qualitative analysis of similarity shows that the prediction for compounds with unknown structure provides promising results using the generated embeddings.

Keywords: Neural Networks · Molecular Similarity · Embeddings.

1 Introduction

Similarity evaluation is widely used in a wide range of applications and fields of research. Recommender systems [15, 6], social networks [17], clustering algorithms [21], and so on take advantage of this. To evaluate similarities, numerical vectors summarize all the available information of the objects to be compared.

Molecular similarity [20, 12] has been extensively used in cheminformatics and related areas such as medicinal chemistry and drug discovery. Its applications include property prediction [5], virtual screening [12], similarity searching [20], and the design of metabolic pathways through the use of compounds molecular structure to guide the search [11, 14]. In practice, molecular descriptors called fingerprints are used to calculate the similarity between compounds. These fingerprints are created from the structures present in the compounds [7]. However, if molecular structure is not available, similarity calculation cannot be performed.

Embedding techniques are a family of methods able to represent objects and their components as numerical vectors. There are two well-known approaches

for graph embeddings. One method is based on neural networks to generate embeddings. An example of this approach is the case of Structural Deep Network Embedding [19]. The other approach uses random walks to travel the graph and describe its components from this information. It builds a representation of each node by sampling its neighborhood through random walks, and combining information of the retrieved paths. This is the case of DeepWalk [13], and in particular Node2Vec [8], which is a modification of the previous one. It has hyperparameters to control the importance between a microscopic view around each node, and a more general view of the whole graph. Based on these ideas, if a metabolic pathway is modeled as a compound graph, its elements can be described as vertex embeddings according to the information of their neighborhoods. This, allow to represent compounds without needing their molecular structures, since compounds embeddings could be constructed directly from the graph structure.

Neural networks have demonstrated an incredible ability to predict physical and chemical properties [16] in several research fields. In a previous work [4], we successfully tested the ability of a Multi-Layer Perceptron (MLP) to predict the similarity between compounds with known structure. We now propose to extend this work by predicting similarity also for compounds with unknown structures, for witch traditional approaches cannot be applied. In fact, we have no knowledge of other methods that could be able to calculate similarity between compounds without having the molecular structures of both compounds.

This work is organized as follow. In Section 2, we describe the algorithm used to build embeddings, the neural model, and how the dataset was created. In Section 3 we present the experiments and their results. Finally, we present conclusions in section 4.

2 Materials and methods

This section presents the theoretical concepts and materials used in this work. First, we describe the embedding algorithm. Then, we show the neural model used and its components finally, we describe the process to build the dataset.

2.1 Embeddings construction

In order to build the embeddings, the Node2Vec algorithm was used. This is a graph-based algorithm that generates embedding by condensing the neighborhood information of each node. It is a supervised method that combines two classical and opposite sampling strategies: Breadth-First Sampling (BFS), that restrict the neighborhood of each node to those which are immediate neighbors of the source one; and Depth-First Sampling (DFS) that takes the neighborhood from sequentially samples of nodes, increasing the distances from the source node. Node2Vec uses randomized walks to incorporate information from a smooth interpolation between BFS and DFS. The balance between both methods is controlled by these hyperparameters, and must be adjusted according to each specific application.

To generate embeddings, this method performs a number of random walks (*numwalk*) starting from each node of the graph. In each step, the information of the neighborhood is combined, and nodes that are taken into account for this process depend of the size of the window (*window*) considered. Given a source node, the algorithm chooses the next one based on hyperparameters p and q, witch control the probabilities to select a new node or to return to a previously selected node. Finally, the information collected from all steps (*lenwalk*) is used to generate an embedding of a specified size (*size*), by using an approach based on Word2Vec [8]. The algorithm has six important tunable hyperparameters that need to be optimized, in order to adequately characterize the compounds and calculate similarity between them.

2.2 Neural model

To perform similarity prediction, a multilayer perceptron [9] (MLP) was trained. As shown in Figure 1, input x corresponds to a couple of concatenated embeddings, one for each compound to be compared, and their similarity as a target output y, is the similarity between considered compounds. In this model, the activation function used in the hidden layers is the ReLU, defined as:

$$ReLU(x) = max(0, x), \tag{1}$$

where x is the linear output of each layer, and max is a function that takes two arguments and return the biggest one. The output layer uses the sigmoid function, defined as:

$$Sigmoid(x) = \frac{1}{1 + e^{-x}}, \tag{2}$$

where x is the linear output of the layer. This function is used to scale the output of the model to values between 0 and 1. Dropout [18] was used as a regularizer with a unique probability p_d for all hidden layers.

We also explored a variable number of hidden layers. According to this description, the neural model has six tunable hyperparameters to be explored: the size of layers 1, 2 and 3; the dropout probability; the learning rate; and the batch size.

2.3 Dataset construction

For this study, two datasets were prepared. Figure 2 shows how the dataset building process was carried out. The entire glycolysis metabolic pathway[1] was used. This pathway is composed of 52 reactions, involving a total of 60 compounds n, of which only 47 have known structure. Data were extracted from KEGG[2] (v95.2). For each compound, the molecular structure was downloaded in SMILES

[1] `https://www.genome.jp/pathway/map00010`
[2] `https://www.genome.jp/kegg/`

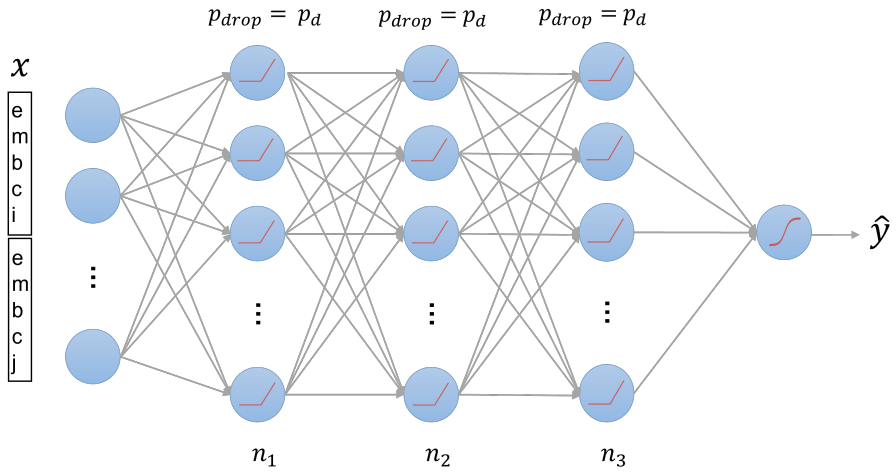

Fig. 1: Model architecture. embci and embcj corresponds to the concatenation of embeddings for compound i and j, respectively.

format, when available from PubChem[3] database. Through the RDKit[4] library was used to calculate MACC keys [7] fingerprints for all the compounds with know structure. Then, Tanimoto coefficient [1] was used to calculate similarity between compounds, according to:

$$T(\mathbf{c}_i, \mathbf{c}_j) = \frac{\sum_k (\mathbf{c}_i^k \wedge \mathbf{c}_j^k)}{\sum_k (\mathbf{c}_i^k \vee \mathbf{c}_j^k)} \tag{3}$$

where $\wedge$ and $\vee$ are the binary operators *and* and *or*, respectively, and $\mathbf{c}_i$ and $\mathbf{c}_j$ are binary representations of the structures of compounds i and j. The Tanimoto coefficient takes values in the range $[0, 1]$ and calculates the proportion of shared features between the two structures.

In total, 1081 patterns were defined resulting from the pairwise combinatorics of the 47 compounds with known structure. As shown in Figure 3, each x input to the neural model results from the concatenation of embeddings for both compounds between which similarity is to be calculated.

3 Results

Creation of embeddings is not a trivial task; both, the algorithm for constructing embeddings from the graph and the neural model have several hyperparameters that must be tuned. Since the embedding algorithm involves six hyperparameters, and the neural network six more, the number of possible combinations to

[3] `https://pubchem.ncbi.nlm.nih.gov/`
[4] `https://www.rdkit.org`

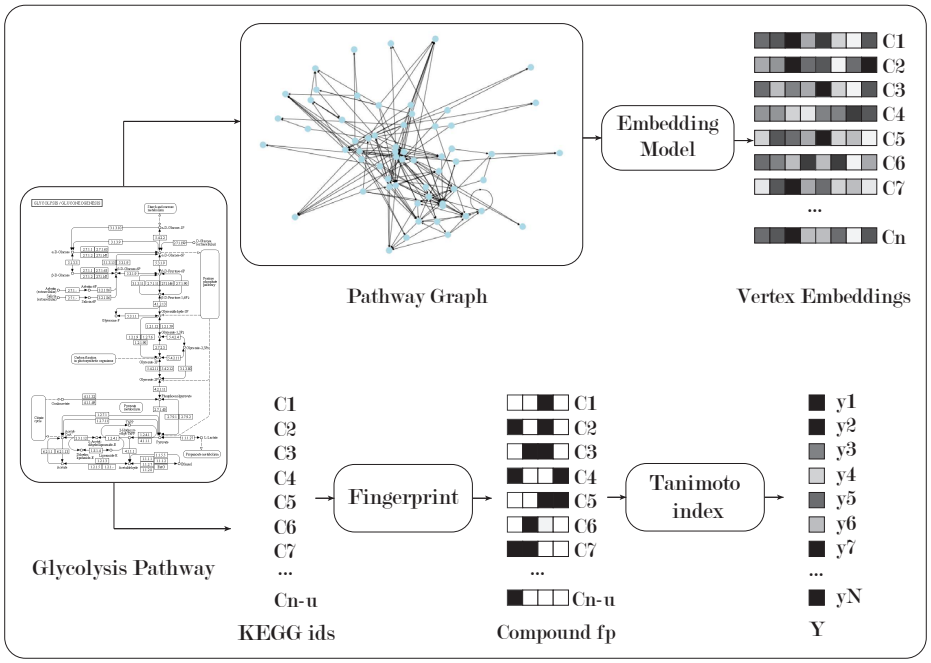

Fig. 2: Dataset pipeline. Where n is the total number of compounds in the Glycolysis pathway, u is the number of unknown structure compounds, and N is the pairwise combinatorics of the compounds with known structure $\frac{(n-u)!}{2!(n-u-2)!} = 1081$.

explore is too large. To avoid this, we split the search in two stages. The first one was carried out to perform an exploration on the hyperparameters for the embedding algorithm. In order to determine if the embeddings were correctly capturing the information needed to characterize compounds, a neural network model with a manually defined set of hyperparameters was trained. The second step was the neural network hyperparameters exploration. It was also done in two steps. Initially, a grid search was performed using the best five embeddings previously found to calculate evaluate the performance. Then, the embeddings and the best hyperparameters found were keep fixed and the hidden layers number was explored.

Clearly, this two-step procedure allows an important reduction in the number of experiments to perform, specially if it is compared with the number required for a full grid search considering all the hyperparameters involved. Thus, although suboptimal configuration could be obtained, the number of experiments is reduced from hundreds of thousands to thousands.

sinc(i) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
E. Borzone, L. Di Persia & M. Gerard; "Neural model-based similarity prediction for compounds with unknown structures"
Applied Informatics 5th International Conference, ICAI 2022, 2022.

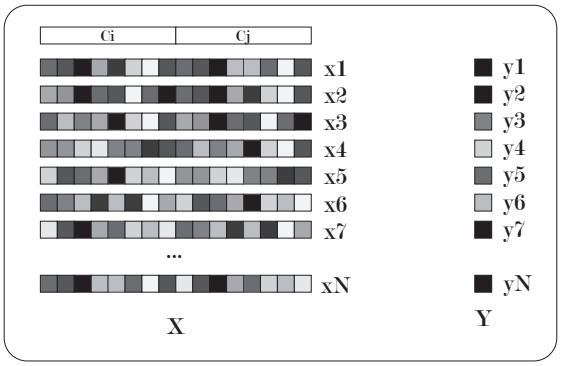

Fig. 3: Final structure of the dataset for the neural network.

3.1 Embeddings selection

Since the embedding algorithm used here is designed to works with any kind of graph, appropriate hyperparameters must be determined for producing embeddings that can be used to correctly predict the Tanimoto index. To determine them, a search was performed. We used the Optuna framework with Tree-structured Parzen Estimator sampler[5] (TPE) [2, 3]. A total of 2,500 combinations of hyperparameters were analyzed: hyperparameters q and p between 0.1 and 1 with a step of 0.1; embedding $size$ = {47, 64, 128, 256}; $window$ sizes = {2, 5, 10, 15}; $numwalk$ between 100 and 500, with a step of 100, and $lenwalk$ between 5 and 20, with a step of 5 were explored. To determine the quality of each embedding, an MLP with architecture [2*$size$,145,20,1] was trained using cross-validation with 5 folds, and a validation portion of 10%. The Mean Square Error (MSE) was used as loss function. Table 1 shows the best five combinations of hyperparameters resulting from the search. The square root of MSE (RMSE) was included in the table since it has the same magnitude order as the Tanimoto index.

q	p	$lenwalk$	$numwalk$	$size$	$window$	RMSE
1	**0.8**	**10**	**300**	**47**	**10**	**0.1124**
1	0.6	10	300	64	15	0.1125
0.8	0.4	10	300	64	2	0.1126
0.9	0.5	20	300	47	5	0.1131
0.9	0.2	15	300	128	5	0.1131

Table 1: Hyperparameter values corresponding to the five embeddings with lower cross-validation error.

[5] https://optuna.readthedocs.io/en/stable/reference/generated/optuna.samplers.TPESampler.html

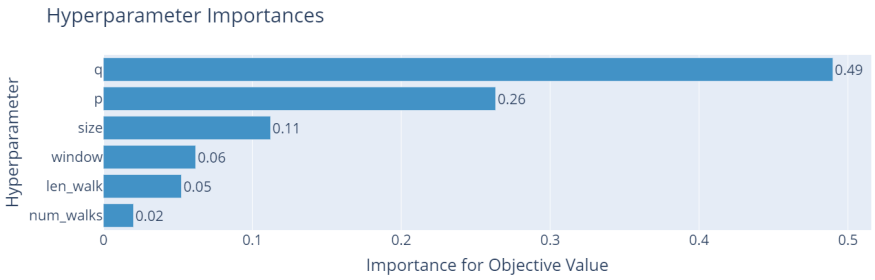

Fig. 4: Embedding algorithm hyperparameter importance analysis

The effect of the embedding algorithm's hyperparameters over similarity prediction was analyzed using FAnova [10]. Results of are presented in Figure 4. As it can seen, the hyperparameter q is the most important in terms of influence on the model error, followed by the hyperparameter p.

3.2 Optimization of the neural architecture

In this section we seek to optimize the architecture of the neural network. Two MLP architectures were considered: one involving 3 hidden layers and another including 4 hidden layers. The experiments were carried out using 6000 training epochs, and 1000 epochs for early stopping. Cross-validation with 5 folds and a validation percentage of 10% was used. The model was implemented using Pytorch 1.9. Training was performed using Adam optimizer, and MSE as cost function.

Since there are multiple hyperparameters that can affect the performance of the model, two grid search were carried out. The first was done with the hyperparameters listed in Table 2. Furthermore, the five best embeddings from the previous experiment were taken as an additional search hyperparameter. Combinatorics performed between the hyperparameters of the Table 3 and each set of embedding. The second grid search was a fine tunning of the layers sizes with the best embedding, learning rate, and batch size found in the previous experiment.

Hyperparamer	Range
Learning rate	$1 \cdot 10^{-5}$, $3 \cdot 10^{-4}$ and $1 \cdot 10^{-3}$
Batch size	10, 50 and 100
P dropout	0.2,0.3 and 0.5
Layer 2 size	120, 145 and 180
Layer 3 size	20, 30 and 50
Layer 4 size	0, 10, 20, 80

Table 2: Hyperparameters explored during architecture optimization.

sinc(i) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
E. Borzone, L. Di Persia & M. Gerard; "Neural model-based similarity prediction for compounds with unknown structures"
Applied Informatics 5th International Conference, ICAI 2022, 2022.

As a result of the grid search, learning rate of $3 \cdot 10^{-4}$, batch size of 100, and dropout probability of 0.3 were identified as the best hyperparameters. We also selected and fixed the best embeddings, which provided the best predictions, being found in 4 of the top 5 models.

Once hyperparameters were chosen, a second grid search was carried out only considering layers sizes. Table 3 shows the search space explored.

Hyperparamer	Range	Step
Layer 2 size	100 - 200	5
Layer 3 size	50 - 200	5
Layer 4 size	50 - 200	5

Table 3: Range of values used in the grid search for hyperparameter exploration.

Layer2	Layer3	Layer4	RMSE
195	**110**	**110**	**0.0928**
185	195	190	0.0932
200	200	150	0.0944
200	185	75	0.0950
185	105	190	0.0955

Table 4: Error obtained in the cross-validation for the selection of neural architecture hyperparameters.

Table 4 shows the five best results of the grid search. The best model was obtained with the following hyperparameters set: learning rate of $5 \cdot 10^{-4}$, batch size of 100, dropout probability of 0.3, and [128, 195, 110, 110, 1] neurons in each layer respectively. The cross-validation $RMSE$ was 0.0928 and, the Pearson coefficient R^2 was 0.92 for the test set. These results indicate a good performance of our model, given that the average absolute error was 9.2%, and the distribution of real and predicted similarity values among compounds was similar. Figure 5 shows the real Tanimoto indices (targets) compared to those predicted by in the neural network one of the best folds. Additionally, trend lines were added for each partition set. We can observe that most of the points are located on the 45° line for all sets, indicating that our model performs good predictions. Futhermore, it can be seen that real and predicted Tanimoto values are similar.

3.3 Similarity prediction in compounds of unknown structure

The embedding method we used does not require the molecular structure of compounds, since it uses the metabolic graph structure to generate compound embeddings. That allow the embedding construction for all compounds, even

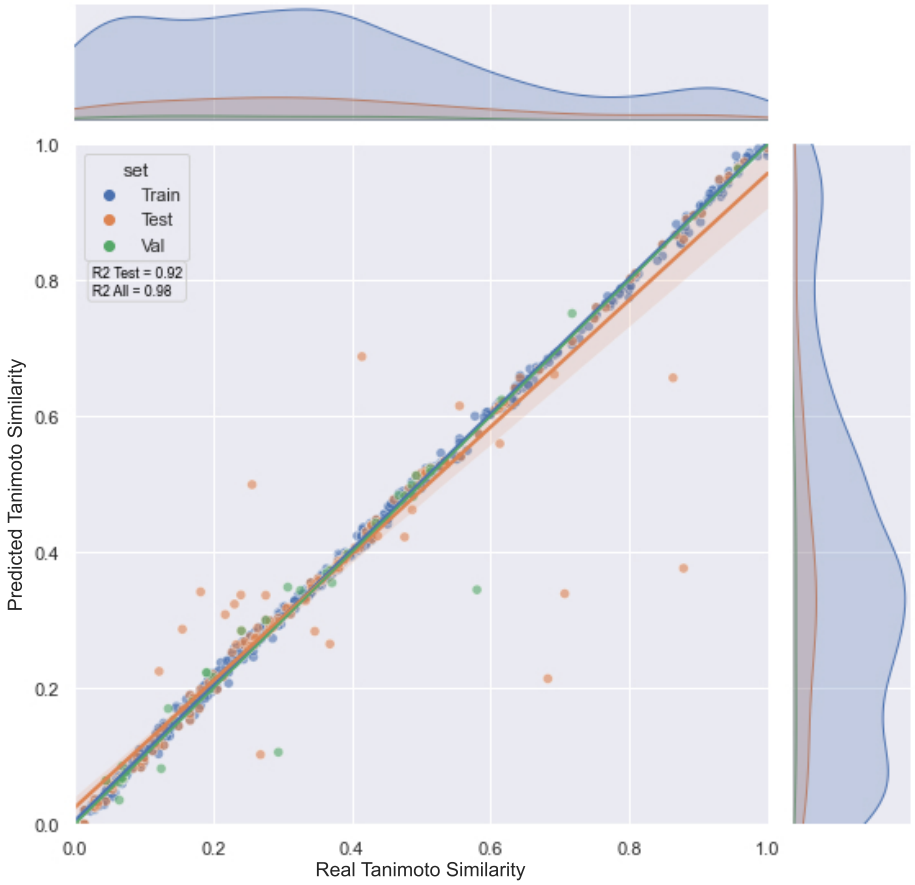

Fig. 5: Real vs predicted Tanimoto similarity are plotted above. R2 indicates to the square correlation coefficient of Pearson for the test set and for all the dataset, respectively. The training values are shown in blue, the test values in orange and the validation values in green.

those for which molecular structure is unknown. To evaluate the quality of the results, a principal component analysis (PCA) for the best embeddings was performed. Figure 6 shows the projection of the embeddings in the first 2 principal component directions. In particular, the compounds belonging to the reaction R03270: C05125 + C15972 $\longleftrightarrow$ C16255 + C00068 are marked with green, and those belonging to the reaction R07618: C15973 + C00003 $\longleftrightarrow$ C15972 + C00004 + C00080 with red. It can be seen how the compounds participating in each reaction are actually close to each other. This is most evident in reaction R03270.

Figure 7 shows the PCA in the area around the reaction R07618. Despite molecular structure of compounds C15972 and C15973 is clearly known, note this includes generic substituent "-R" in both cases. In consequence, similarity

sinc(i) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
E. Borzone, L. Di Persia & M. Gerard; "Neural model-based similarity prediction for compounds with unknown structures"
Applied Informatics 5th International Conference, ICAI 2022, 2022.

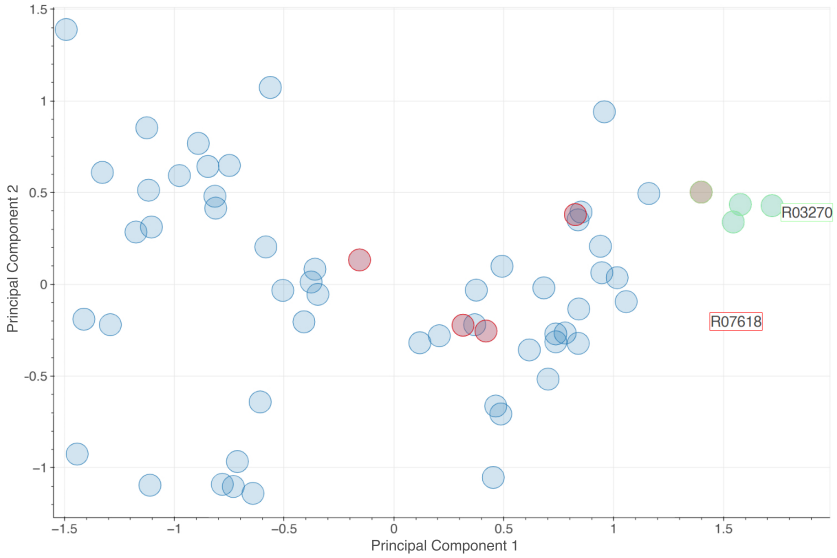

Fig. 6: Graph of the first two principal components for embeddings. It can be seen that the compounds involved in each reaction appear close to each other on the graph.

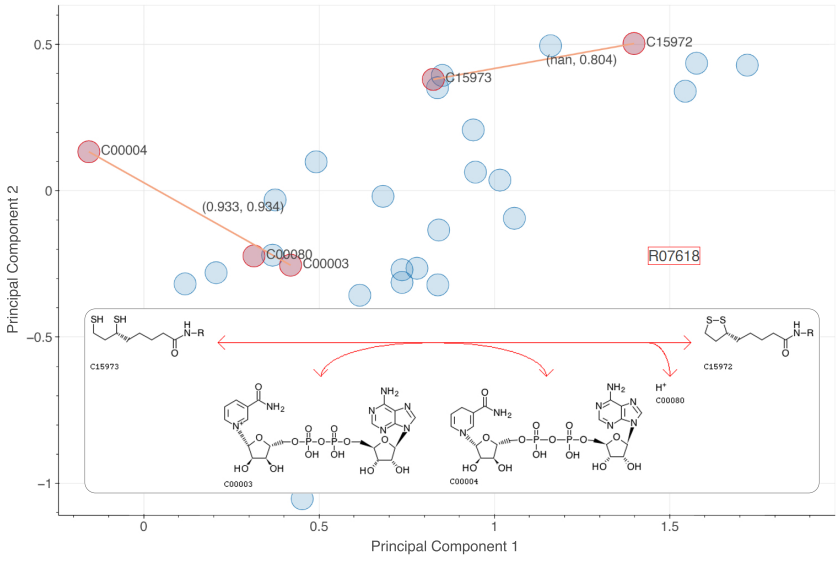

Fig. 7: Enlargement of a region of the figure around reaction R07618. A yellow line joins certain selected compounds to be compared. On the left are shown the target values and on the right the values predicted by the neural model.

sinc(i) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
E. Borzone, L. Di Persia & M. Gerard; "Neural model-based similarity prediction for compounds with unknown structures"
Applied Informatics 5th International Conference, ICAI 2022, 2022.

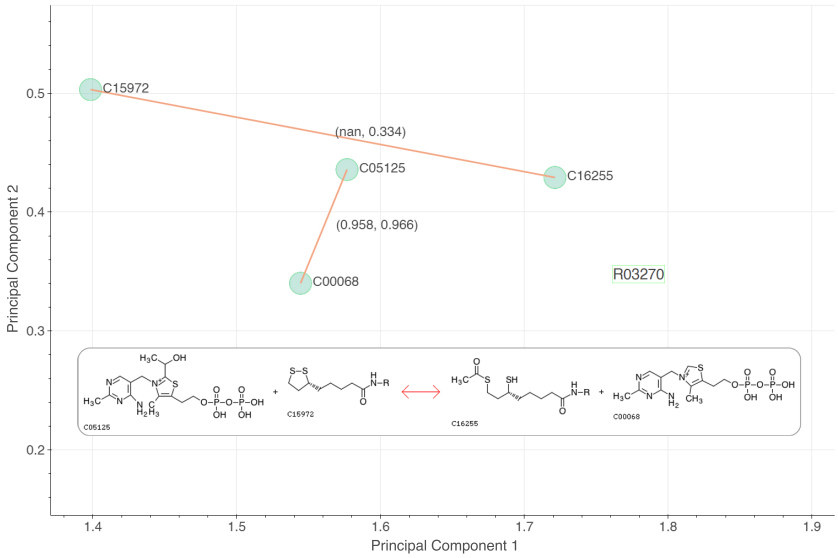

Fig. 8: Enlargement of a region of the figure around reaction R03270. A yellow line joins certain selected compounds to be compared. On the left are shown the target values and on the right the values predicted by the neural model.

can be only subjectively estimated since this information cannot be processed to build fingerprints. This situation does not affect our approach, since it uses embeddings learned from the graph topology of the metabolic network to represent compounds. As a result, predicted similarity (0.804) is in accordance wit visual analysis, since both compounds share a high proportion of their molecular structures. Furthermore, predicted similarity for compounds C00004 and C00003 is 0.934, that is in accordance with the real value (0.933).

Figure 8 shows an excerpt of the PCA projection, in the area around reaction R03270. This reaction involves the compounds C15972 and C16255 with no known structure. This reaction decomposes the substrates C05125 and C15972 into the products C16255 and C00068. As expected, the similarity between C05125 and C00068 shows high similarity (real: 0.958, predicted: 0.966) since they share a large part of their structures. On the other hand, predicted similarity for the C15972 and C16255 is 0.334, indicating that they do not share an important proportion of their structures. However, we can appreciate from structures in chemical equation that it is not correct. Examples like this indicate that more information needs to be incorporated into the model in order to get more accurate predictions for unknown structures.

sinc(i) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
E. Borzone, L. Di Persia & M. Gerard; "Neural model-based similarity prediction for compounds with unknown structures"
Applied Informatics 5th International Conference, ICAI 2022, 2022.

4 Conclusions

This work addresses the problem of predicting similarity between compounds when molecular structure is not available. For this purpose, a metabolic pathway was modeled as a graph, where each node is a compound and the edges connect compounds that are substrates and products of the same chemical reaction. Using a general algorithm to generate node embeddings, descriptors for compounds were generated according to their neighborhoods and used to train a neural model to predict similarity. For the construction of the embeddings, a total of 2500 combinations of hyperparameters of the embedding algorithm were explored. From this search, the best 5 were selected to explore together with these the hyperparameters of the MLP through a grid search. Finally, the best set of embeddings and the hyperparameters that gave the best results were selected to explore the network architecture through another grid search. The results clearly shows that it is possible to predict similarity between compounds using neural networks and the proposed embedding strategy. Moreover, this approach allows to obtain an average error close to 10%. The proposal is expected to be improved by incorporating more information, in order to generate more robust descriptors to predict the similarity between compounds.

References

1. Bajusz, D., Rácz, A., Héberger, K.: Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? Journal of Cheminformatics **7**(1), 20 (Dec 2015). https://doi.org/10.1186/s13321-015-0069-3
2. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for Hyper-Parameter Optimization. In: Advances in Neural Information Processing Systems. vol. 24. Curran Associates, Inc. (2011)
3. Bergstra, J., Yamins, D., Cox, D.: Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In: Dasgupta, S., McAllester, D. (eds.) Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 28, pp. 115–123. PMLR, Atlanta, Georgia, USA (17–19 Jun 2013)
4. Borzone Eugenio, Gerard Matias, D.P.L.: Evaluación de un modelo neuronal para la estimación de similaridad entre compuestos a partir de representaciones one-hot. In: 52st JAIIO Jornadas Argentinas de Informática - ASAI (2022)
5. Brown, R.D., Martin, Y.C.: Use of structure-activity data to compare structure-based clustering methods and descriptors for use in compound selection (1996)
6. Covington, P., Adams, J., Sargin, E.: Deep Neural Networks for YouTube Recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems. pp. 191–198. ACM, Boston Massachusetts USA (Sep 2016). https://doi.org/10.1145/2959100.2959190
7. Durant, J.L., Leland, B.A., Henry, D.R., Nourse, J.G.: Reoptimization of MDL Keys for Use in Drug Discovery. Journal of Chemical Information and Computer Sciences **42**(6), 1273–1280 (Nov 2002). https://doi.org/10.1021/ci010132r
8. Grover, A., Leskovec, J.: node2vec: Scalable Feature Learning for Networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge

Discovery and Data Mining. pp. 855–864. ACM, San Francisco California USA (Aug 2016). https://doi.org/10.1145/2939672.2939754
9. Haykin, S.: Neural networks: a comprehensive foundation. Prentice Hall PTR (1994)
10. Hutter, F., Hoos, H., Leyton-Brown, K.: An efficient approach for assessing hyperparameter importance. In: Xing, E.P., Jebara, T. (eds.) Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 32, pp. 754–762. PMLR, Bejing, China (22–24 Jun 2014)
11. McShan, D.C., Rao, S., Shah, I.: PathMiner: predicting metabolic pathways by heuristic search. Bioinformatics **19**(13), 1692–1698 (1 Sep 2003)
12. Muegge, I., Mukherjee, P.: An overview of molecular fingerprint similarity search in virtual screening. Expert Opinion on Drug Discovery **11**, 137–148 (2 2016). https://doi.org/10.1517/17460441.2016.1117070
13. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: Online Learning of Social Representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710 (Aug 2014). https://doi.org/10.1145/2623330.2623732, arXiv:1403.6652 [cs]
14. 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)
15. Steck, H., Baltrunas, L., Elahi, E., Liang, D., Raimond, Y., Basilico, J.: Deep Learning for Recommender Systems: A Netflix Case Study. AI Magazine **42**(3), 7–18 (Nov 2021). https://doi.org/10.1609/aimag.v42i3.18140, number: 3
16. Thomsen, J.U., Meyer, B.: Pattern recognition of the 1H NMR spectra of sugar alditols using a neural network. Journal of Magnetic Resonance (1969) **84**(1), 212–217 (Aug 1989). https://doi.org/10.1016/0022-2364(89)90021-8
17. Tiwari, S.P.: Social Media Based Recommender System for E- Commerce Platforms. International Journal of Research in Engineering and Science (IJRES) pp. 87–98 (2021)
18. Wager, S., Wang, S., Liang, P.S.: Dropout Training as Adaptive Regularization p. 9 (2013)
19. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 1225–1234. KDD '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939753
20. Willett, P., Barnard, J.M., Downs, G.M.: Chemical similarity searching. Journal of Chemical Information and Computer Sciences **38**, 983–996 (1998). https://doi.org/10.1021/ci9800211
21. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of The 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 478–487. PMLR, New York, New York, USA (20–22 Jun 2016)