Predicting the number of biochemical transformations needed to synthesize a compound

João Correia
Centre of Biological Engineering
University of Minho
Braga, Portugal
LABBELS – Associate Laboratory,
Braga/Guimarães, Portugal
jfscorreia95@gmail.com

Rafael Carreira
SilicoLife, Lda
Braga, Portugal
rcarreira@silicolife.com

Vítor Pereira
Centre of Biological Engineering
University of Minho
Braga, Portugal
LABBELS – Associate Laboratory,
Braga/Guimarães, Portugal
vpereira@ceb.uminho.pt

Miguel Rocha
Centre of Biological Engineering
University of Minho
Braga, Portugal
LABBELS – Associate Laboratory,
Braga/Guimarães, Portugal
mrocha@di.uminho.pt

***Abstract*—Exploiting the natural metabolic abilities of microorganisms for the production of bioactive compounds has been a research problem of great interest. The economical and environmental costs associated with petrochemical-derived industries have promoted the emergence of biochemical processes from renewable carbon sources. However, optimally rewiring microbial metabolism in a competitive and sustainable manner is still a challenge. Recently, some retrobiosynthesis tools for the design of *de novo* biosynthetic pathways have been proposed. These tools generate a large number of intermediate compounds that are beyond experimental feasibility. Thus, effective methods to reduce the number of compounds by selecting the most promising ones are still needed. Here, we propose the use of classification and regression deep learning models, such as fully-connected neural networks and 1D convolutional neural networks, to predict the number of biochemical transformations needed to produce a compound. The data to train and evaluate the models was generated using a set of 13055 reaction rules and 673 compounds from *Escherichia coli* metabolism as starting compounds. The data was generated up to 5 steps resulting in a dataset of over 2.6 million compounds. This approach can be effectively used in biochemical applications, including retrobiosyntesis, to prioritize compounds that can be produced using fewer biochemical transformations.**

***Index Terms*—deep learning, reaction rules, biochemical transformations, biosynthesis**

I. Introduction

During the last decades, microorganisms have been extensively used as a platform for the production of added-value compounds with a wide set of applications in the pharmaceutical, chemical, food, and agriculture industries. It was in 1928 when Sir Alexander Fleming discovered Penicillin from *Penicillium notatum* [1], that microorganisms were seriously regarded as a source of natural products triggering a big wave of interest in the subject. Since then, the production of drugs, chemicals, and biofuels using microorganisms grew exponentially.

Microbial biosynthesis offers many advantages over traditional chemical synthesis. While traditional chemical synthesis demands high energetic resources and often produces toxic intermediates, microbial biosynthesis is safer and ecofriendlier [2]. Additionally, taking into account how developed the fields of metabolic engineering (ME), protein engineering, and synthetic biology are nowadays, the high accessibility of engineered strains to produce specific compounds makes the process of redesigning microbial cellular networks and fine-tuning physiological capabilities much easier [3]. ME also offers established pathway optimization methods for improvements on the yield and productivity of target compounds [4].

While multiple organisms can be used and are optimized to produce specific compounds, none is as well characterized as *Escherichia coli*. *E. coli* has been studied over several decades and has shown its potential in many functional genomics and ME efforts [5], [6]. Moreover, its ability to quickly grow on minimal media, while maintaining its full metabolic function offers an important advantage over other popular organisms, such as the yeast [7].

In the last years, the application of machine learning (ML) and deep learning (DL) in bioinformatics has seen a growing interest as biological data becomes more accessible. In particular, the availability of comprehensive datasets of omics data and metabolic pathways propelled the use of DL within the field of ME. DL is a subfield of ML that uses multi-layer neural networks to learn hierarchical abstract features using a set of non-linear modules that transform, at each step, the original input data into more abstract representations [8]. DL has achieved remarkable results in many fields including computer vision [9], speech recognition [10] and bioinformatics [11]. In particular, the use of DL in MEs is evolving at a fast pace. Some of the most prolific applications include product maximization by, for example, predicting optimal reactor conditions, *de novo* pathway design, phenotypic profiling, and robust system modeling [12].

One recent challenging subject in ME involving the application of DL, in particular reinforcement learning, is retrobiosynthesis. Retrobiosynthesis consists in efficiently breaking a target compound finding a series of producing reactions, until readily available starting materials are obtained. Retro-

biosynthesis is commonly carried on using template-based approaches by identifying a set of reaction rules, representing enzymatic reactions that have been observed in biological systems, that can transform native metabolites of a host into the target molecule [13], [14]. Other approaches do not rely on reaction templates and use DL models to directly predict the outcome of reactions in the reverse order [15], [16]. However, the debate on whether template-based approaches are long-term feasible, due to the sheer number of reaction templates available for matching, or if the role of human experts is indispensable due to the complexity of the reaction mechanism is still on-going [17].

The tree of possibilities generated by retrosynthesis approaches results in a combinatorial explosion of possible pathways. Thus, exhaustive exploration of these pathways is not computationally feasible. To effectively explore the most promising routes, a set of heuristics to prune the tree needs to be defined. Many retrosynthesis studies approach this problem in different manners. For example, in Simpheny [18], to constrain the network size a maximum molecule size is defined. In RetroPath [19], the network complexity is limited by coding substrates, products, and reactions into molecular signatures. A reaction signature is given by the difference between the signatures of products and substrates. This signature can be controlled using predefined distances, defining graph distances of atoms, which leads to different levels of specificity, thus the number of *de novo* pathways can also be controlled. In RetroPath 2.0 [14], they also use signatures, but introduce an additional enzyme score reflecting the ability to retrieve enzyme sequences catalyzing defined transformations. SimZyme/SimIndex [20] and PathPred [21] use similarities between the molecules and the typical substrate of an enzyme.

Despite many studies suggesting different strategies to define which compounds to prioritize when searching for the best pathways to synthesize a compound, there is no single method that is regarded as the best. Another important factor to consider when computing these routes is how easy they are to reproduce in a host microorganism. For this, different factors such as atom conservation, thermodynamics, presence or absence of toxic intermediates, product yield and pathway length are taken into consideration when ranking the obtained pathways.

In this study, we propose the use of DL to predict the number of biochemical transformations needed to synthesize a compound. We tested different DL architectures, in specific, fully connected deep neural networks and 1D convolutional neural networks. We also tested different compound representations, such as molecular fingerprints and transformer-based molecular embeddings. To our knowledge, this is the first attempt to leverage DL to model the number of steps needed to synthesize a compound. To train the DL models, we generated data using reaction rules from public databases with compounds from the *E. coli* metabolism as starting materials. The performance of our models indicates that DL can promote the study of compound synthetisability in host microorganisms and it is expected that it can represent a useful tool to effectively narrow down a large number of retrobiosynthesis-derived compound candidates to more promising routes.

II. Materials and Methods

A. Data

In this work, multiple DL architectures and compound representations were tested. The data used to train the models was generated using a set of 13055 reaction rules and a set of 673 compounds from *E. coli* metabolism as available starting precursor compounds. The reaction rules were collected from two public resources, RetroRules [22] and MINE databases [23].

Reaction rules. Reaction rules are generic descriptions of reactions that encode the way reactants are converted into products. Reaction rules are very important for a variety of synthetic biology approaches, mainly in *de novo* pathway discovery. A reaction rule can be applied to a compound if the compound contains a particular substructure that is encoded by the reaction rule. Then, new product compounds can be generated by applying the transformation encoded by the reaction rule. These rules can encode single and multi reactant reactions and can generate multiple products if they match multiple parts of the reactant molecules. In Fig. 1, an example of a known reaction is shown (id MNXR94682 in the MetaNetX database [24]), which was used to generate a reaction rule.

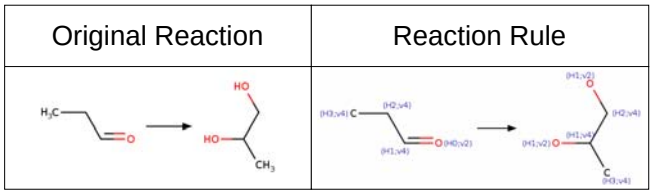

Fig. 1. Example of a reaction rule and respective original reaction.

The reaction rules from RetroRules were downloaded directly from the website https://retrorules.org/dl. Only the reaction rules that apply in the forward direction and with a diameter of 2 were selected resulting in a set of 5732 rules. RetroRules contains reaction rules that were generated using different diameters, in specific, 2 to 16. This diameter consist in the distance considered around the reaction center of the original metabolic reaction when generating the reaction rule. So one reaction can be used to generate multiple reaction rules at different diameter levels. The higher the diameter, the more specific the rules will be. This diameter directly links with the known promiscuity of enzymes. We selected the reaction rules with diameter 2 for these reasons, they are less specific and promote enzyme promiscuity.

The 7323 MINE database reaction rules were downloaded from the GitHub page https://github.com/tyo-nu/MINE-Database and were used as they are. The final dataset consisted of 13055 reaction rules identified using the SMARTS notation [25]. The validity of each reaction rule SMARTS

was determined using RDKit [26], an open source toolkit for cheminformatics.

Starting compounds. The list of starting precursors that we assume to be available are the ones existing in the metabolism from *E. coli*. We selected this microorganism because it is widely used in a great set of biochemical processes including in the synthesis of added-value compounds. These *E. coli* sink compounds were downloaded from the RetroPathRL [27] GitHub https://github.com/brsynth/RetroPathRL. The authors extracted these compounds from the *E. coli* iJO1366 genome-scale metabolic model [28]. They selected the compounds that lie in the cytosol compartment except for the ones that cannot be produced by any reactions in a steady-state metabolic model. This was obtained by performing a Flux Variability Analysis [29] using the COBRApy package [30]. After downloading these data, we selected only the compounds with available and valid identifiers, in this case, the International Chemical Identifier (InChI) [31], resulting in a set of 673 compounds. The validity of each compound InChI was determined using RDKit. For further usability, the compounds were converted into the SMILES notation [32] also using RDKit.

Generated Dataset. The dataset used to train and evaluate our DL models was generated by successively trying to applying randomly selected reaction rules to randomly selected compounds from the previous step starting from the *E. coli* iJO1366 sink compounds (step 0). In the first step, because the number of compounds was low, we applied all possible reaction rules to all *E. coli* iJO1366 sink compounds, resulting in a set of 146157 compounds. After this step, we tried to apply randomly selected reaction rules to randomly selected compounds over 5 million iterations. This second step resulted in a set of 464994 compounds. We repeated this process up to five steps, resulting in the dataset described in Table I. When a compound that was already generated in a previous step was generated again, only the first one was kept. We defined a maximum of five biochemical transformations based on the fact that, in general, microbial networks have small-world properties and thus small average path length [33], [34]. This means that, in theory, with a small number of biochemical transformations we can go from any compound to any compound in the network.

TABLE I
NEW COMPOUNDS GENERATED AT EACH STEP.

Step	Generated Dataset	Independent Dataset
1	146157	16439
2	464994	27151
3	600280	44681
4	698529	97249
5	773586	70342
Total	2683546	255862

After the dataset was generated it was divided into train, validation, and test sets using ScikitLearn [35]. The splits were made in a stratified fashion based on the step value with the proportions of approximately 60%, 20%, and 20% for train, validation, and test sets respectively.

Independent Dataset. Since the compounds present in later steps were generated using the compounds from the previous step, there is a dependency between the compounds generated at each step. To validate if the previously generated dataset was representative enough we generated an independent set using the same approach. The newly generated independent test set comprises 255862 unique compounds. The number of compounds for each step is shown in Table I.

B. Molecular Representations

There has been a lot of research on how to better represent compounds in a suitable form so that ML algorithms can learn and generalize the information shared among sets of compounds. Several approaches to encode the properties and structural characteristics of compounds have been reported in the literature. From single descriptors to complex multi-dimensional graph-based formulations, compound representation remains a hot topic. The most traditional molecular features include molecular descriptors and fingerprints, such as the Extended-Connectivity Fingerprints [36]. Other successful approaches that make use of representations like other types of fingerprints, weave, graph convolutions, and Natural Language Processing (NLP)-inspired embeddings are actively being used [37]–[39]. In this study, we focus on two distinct molecular representations, the well-known Morgan fingerprints [36] and the NLP-based Molecular Transformer Embeddings [40].

Morgan Fingerprints. Also known as circular fingerprints, Morgan fingerprints are built by applying the Morgan algorithm to a defined set of atom invariants up to a defined radius around each atom of the molecule. We computed Morgan fingerprints of radius 2 hashed to 1024 bits using the RDKit package.

Molecular Transformer Embeddings (MTE). With the emergence of the transformer architecture [41], the field of NLP saw some considerable improvements. The use of such architecture has been explored in the field of chemoinformatics, in particular by translating between two distinct text-based molecular representations in well-studied subsets of the chemical space. In the study by Morris *et al.* [40], the authors trained and repurposed, through transfer learning, a transformer network to predict binding affinity. The intermediate set of features representing abstract features that describe general molecular structures that are generated by this architecture can be used as embeddings. These embeddings can then be used as features to train other models for diverse purposes. We computed these MTE for our datasets. We defined a maximum length of our compound SMILES of 300 characters and an embedding size of 512.

C. Deep Learning Models

In this study, we leverage the use of DL models to predict the number of biochemical transformations needed to synthesize a compound. We defined this problem as both a multiclass classification problem with 5 labels, the 5 steps, and as a

regression problem, where the number of biochemical steps is modulated as a continuous variable. For this, we use fully connected neural networks (FCNNs) and 1D convolutional neural networks (CNNs) for both classification and regression tasks.

Fully Connected Neural Networks. A FCNN consists in a set of fully connected layers. In a fully connected layer, all possible neuron connections between layers are present, meaning that every input dimension influences each output dimension. Fully connected layers are the most common layers in artificial neural networks. One of the major advantages of FCNNs is that no special assumptions about the input data need to be made, making them very broadly applicable. However, these networks are very computationally intense, prone to overfitting, and tend to have weaker performance than specific networks tuned for the problem in question.

In this study, we used FCNNs for both classification and regression tasks using Morgan fingerprints and MTE as inputs. All our FCNN models follow a similar architecture. The models consists of an initial input Dense layer with 1024 units in the case of using Morgan fingerprints and 512 for the MTE. Then a variable number of Dense, with L1 and L2 regularizers, BatchNormalization and Dropout layers are added to the model. A final output Dense layer concludes the model. The initial and intermediate Dense layers use the Relu activation function. In the case of classification, the final dense layer has 5 units and uses softmax as the activation function. In this case, we also use the categorical cross-entropy as the loss function and accuracy as the metric to follow during training. In the case of regression, the last Dense layer has 1 unit and uses the linear function as activation function. The loss function and metric to follow during training was the mean absolute error. We used the Adam as our optimization algorithm and a ReduceLROnPlateau callback with a minimum learning rate of 0.000001, a factor of 0.25, and patience of 10 epochs that reduces the learning rate by 25% if there is no improvement in the validation loss for 10 epochs. We defined a maximum of 250 epochs and a batch size of 256. We also use the EarlyStopping callback with patience of 15 epochs that stops training when there is no improvements in the validation loss for 15 epochs.

1D Convolutional Neural Networks. CNNs are a class of feed-forward artificial neural networks that gained popularity during the last decades and became the dominant method for computer vision applications. CNNs employ mathematical operations called convolutions in at least one of its layers. Other building blocks of CNNs are pooling and fully connected layers. Both convolution and pooling layers are designed to identify spatial hierarchies of features that are then mapped to the final output by the fully connected layers. Despite being designed for two-dimensional inputs, CNNs can also deal with other dimensional inputs such as 1D and 3D. As opposed to 2D convolutions that act in two dimensions, 1D convolutions operate in only one dimension applying 1D convolutions (scalar multiplications and additions). This allows the use of these types of models in data such as 1D numerical vectors, including molecular descriptors, fingerprints, and embeddings.

In this study, we used 1D CNNs for both classification and regression tasks using Morgan fingerprints and MTE. All our 1D CNN models follow a similar architecture. The models consist of an initial GaussianNoise layer that adds noise to the data helping to mitigate overfitting, followed by 2 Conv1D layers using the Relu activation function. Then one Flatten and one Dropout layer are added to the model. Before the output layer, a set of 1 Dense, 1 BatchNormalization, and 1 Dropout layers are added to the model. The model is then concluded with an output Dense layer. We used Adam as our optimization algorithm and a ReduceLROnPlateau callback with a minimum learning rate of 0.000001 with a factor of 0.25 with patience of 10 epochs. We defined a maximum of 150 epochs and a batch size of 512. We also used the EarlyStopping callback with patience of 15 epochs.

D. Performance metrics

The use of performance metrics is a key step in any ML pipeline to ensure that the model is performing as expected. There are dozens of metrics for both classification and regression, we will discuss the ones used in this study.

Classification metrics: Some of the most popular metrics in classification tasks include classification accuracy, precision, recall, F1-score, and the confusion matrix, which is essential to compute some metrics.

- Confusion Matrix: The confusion matrix is a two-dimensional matrix that summarizes the classification performance of a classifier regarding the model label predictions versus ground-truth labels [42]. Each row in the confusion matrix represents the predicted classes by the classifier and each column represents the actual class. In a confusion matrix for a binary classification there are four important terms: True positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs). TPs represent the number of positive class samples that were predicted correctly. TNs represent the number of negative class samples that were predicted correctly. FPs represent the number of negative class samples that were predicted incorrectly. FNs represent the number of positive class samples that were predicted incorrectly. In multiclass problems there are no negative or positive classes, so the TPs, TNs, FPs, and FNs are calculated for each class.
- Accuracy: This classification metric represents the ratio of correct predictions in the total number of input samples [43].
- Precision: Focused on Type-I errors, precision measures the fraction of positive class predictions that were actually positive [44]. Precision for each class can be calculated using the following formula: $\frac{TPs}{TPs+FPs}$. Values of precision near 1 mean that the model is performing well on classifying the positive cases as positive, whereas low precision scores means that the model is classifying a high portion of negative cases as positive, i.e. produces a high number of false positives.

- Recall: Focused on Type-II errors, recall, also known as sensitivity, measures the fraction of all true positive samples that were actually predicted as positive by the model [45]. Recall for each class can be calculated using the following formula: $\frac{TPs}{TPs+FNs}$. Recall near 1 mean that the model is not missing many TPs, i.e. can correctly classify positive samples as positive. Low recall values mean that a high number of FNs are being predicted by the model, i.e. the model is classifying positive samples as negative.
- F1-score: This metric combines both precision and recall, actually it is the harmonic mean of these two metrics [46]. The f1-score can be calculated using the following formula: $\frac{2}{\frac{1}{\text{precision}}+\frac{1}{\text{recall}}} = \frac{2TPs}{2TPs+FPs+FNs}$. In a perfect model, the respective f1-score would be equal to 1, which means that both precision and recall would also be equal to 1 which also means that the model would have a 100% accuracy. The f1-score is particularly important on imbalanced classification problems as it tells you how precise your model is, i.e. how many cases it classifies correctly, and how robust it is, i.e. if it misses a significant number of samples.

Regression metrics: The output of regression models are continuous values. So, we need metrics that can calculate the numeric distance between the predicted value and the ground truth. Some of the most popular metrics in regression tasks include Mean Absolute Error (MAE), Mean Squared Error (MSE) and, R^2.

- MAE: This metric computes the average difference between the predicted values and the ground truth values [47]. Mathematically, it can be represented as: $\frac{1}{N}\sum_{j=1}^{N}|y_j - \check{y}_j|$, where y_j is the ground truth value, $\check{y}_j$ is the predicted value and N is the number of instances [47]. MAE gives a measure of how far the predictions are from the actual value. However, it does not give the direction of the error, i.e. if we are under-prediction or over-predicting the data.
- MSE: Perhaps the most popular metric in regression problems, MSE computes the average of the squared difference between the predicted values and the ground truth values [48]. Mathematically, it is represented by the following formula: $\frac{1}{N}\sum_{j=1}^{N}(y_j - \check{y}_j)^2$, where y_j is the ground truth value, $\check{y}_j$ is the predicted value and N the number of instances [48]. When compared with MAE it penalizes outliers harder by squaring them. However, this sometimes can lead to an overestimation of how bad the model is.
- R^2: Also known as the coefficient of determination, the R^2 calculates the proportion of variance which is explained by the predictor variables in the sample [49]. When the R^2 is close to 1, it means that the model was able to capture a high proportion of the variance in the target variable. If it is close to 0 it means that the model wasn't able to capture any variance in the target variable.

E. Development Environment

This work was developed using Python version 3.6.12. Molecular operations like compound SMILES validity, standardization, reaction SMARTS validity, and Morgan fingerprints generation was done using RDKit version 2019.09.3. Molecular Transformer Embeddings were generated using the release from September 16, 2020. All the models were implemented using Tensorflow version 2.2.0. Hyperparameter optimization and model training and evaluation were done using scikit-learn version 0.23.2. Source code and small data examples are available at https://github.com/jcorreia11/WCCI2022_scripts.

III. Results and Discussion

Before training any DL model, we divided our data into three sets, training set, validation set, and test set. We made sure that there was no duplicated data in and between all our datasets. Only the train and validation sets were used in the parameter optimization and training phases. The models only saw the test set and the independent set in the evaluation phase to produce the metrics shown here.

We performed 5-fold hyperparameter optimization using RandomizedSearchCV for 15 iterations for all our FCNN models. We optimized the number of hidden dense layers, the number of units in the hidden layers, the first dropout, the dropouts that followed hidden layers, and the l1 and l2 regularizers. In Table II, the optimized parameters and the ones that produced the best results for each case are shown. For our 1D CNN models, we performed a 3-fold hyperparameter optimization using RandomizedSearchCV for 10 iterations. We optimized the standard deviation of an initial GaussianNoise layer, the number of output filters and kernel size of the Conv1D layers, the number of units in the Dense layers, and the dropout ratio introduced by the Dropout layers. In Table III, the optimized parameters and the ones that produced the best results for each case are shown.

In Fig. 2, the accuracy for the four classification models using the test and independent sets are shown. The use of Morgan fingerprints to encode the compounds was considerably superior when compared with the embeddings generated with the MTE. The use of FCNNs versus 1D CNNs produced better results, but the differences were smaller than the differences obtained when comparing the molecular representation methods. The FCNNs with Morgan fingerprints obtained a test accuracy of 63% which was 4% better than our second best performing model, the 1D CNN also with Morgan fingerprints.

The best classification model, the FCNN with Morgan fingerprints, was trained using the best parameters from the hyperparameter optimization search for 27 epochs (early stopping with a maximum of 250 epochs) and obtained a training accuracy of 0.735, a validation accuracy of 0.646, and a test accuracy of 0.630.

By analyzing the results shown in Table IV, we can see that the model performs the best in classifying compounds that require 1, 2, and 5 biochemical transformations to be synthesized with higher precision, recall, and f1-score metrics.

TABLE II
PARAMETERS OPTIMIZED USING A 5-FOLD RANDOMIZEDSEARCH FOR THE FCNNS.

Parameter	Values	Morgan Classification	MTE Classification	Morgan Regression	MTE Regression
# of hidden layers	2, 4, 6	2	2	6	2
Hidden layers units	1024, 512, 256	512	1024	256	512
First dropout	0, 0.2, 0.5	0.2	0	0.2	0
Dropout hidden layers	0, 0.3, 0.4	0	0.4	0	0.3
l1	0, 0.001, 0.01	0	0	0	0
l2	0, 0.001, 0.01	0	0.01	0	0

TABLE III
PARAMETERS OPTIMIZED USING A 3-FOLD RANDOMIZEDSEARCHCV FOR THE 1D CNNS.

Parameter	Values	Morgan Classification	MTE Classification	Morgan Regression	MTE Regression
Gaussian noise stddev	0.01, 0.05	0.05	0.01	0.05	0.05
Size of output filters	4, 8, 16	16	8	16	8
Kernel size	32, 64, 128	32	32	64	64
Dense layers units	512, 256, 128	512	512	256	128
Dropout	0, 0.3, 0.5	0.5	0.3	0.5	0

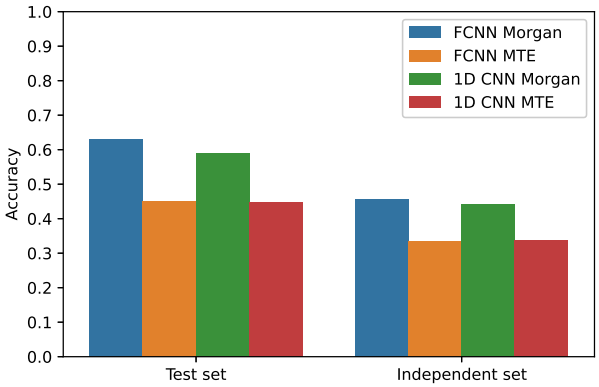

Fig. 2. Test and independent set accuracy for all models.

The compounds that require 3 and especially 4 biochemical transformations have worse performance metrics. Additionally,

TABLE IV
CLASSIFICATION REPORT OF THE FCNN WITH MORGAN FINGERPRINTS.

Step	Precision	Recall	F1-Score
1	0.77	0.81	0.79
2	0.61	0.79	0.69
3	0.52	0.55	0.54
4	0.49	0.37	0.42
5	0.66	0.65	0.66

if we take a closer look at the confusion matrix in Table V we can see that the majority of the mispredictions, around 78%, only fail by one step, which may be a reasonable estimate in practical applications since we are looking for an estimation of how near we are from our available starting precursors. This can also mean that this problem can better be modeled as a regression task. If we consider that our model predictions are correct if the output step is no more that 1 step, up and

TABLE V
CONFUSION MATRIX OF THE FCNN WITH MORGAN FINGERPRINTS.

Step	1	2	3	4	5
1	25141	3316	228	101	82
2	4073	77539	9361	1632	753
3	1229	21039	76363	19234	2115
4	841	10837	30449	75675	22091
5	594	7145	17113	46150	83609

down, far from the labeled value, our accuracies increase by a considerable margin achieving a 92% test and 91% independent test accuracies using the FCNN with Morgan fingerprints. The test and independent test accuracies, when allowing a 1 step margin error, for all our models are shown in Fig. 3.

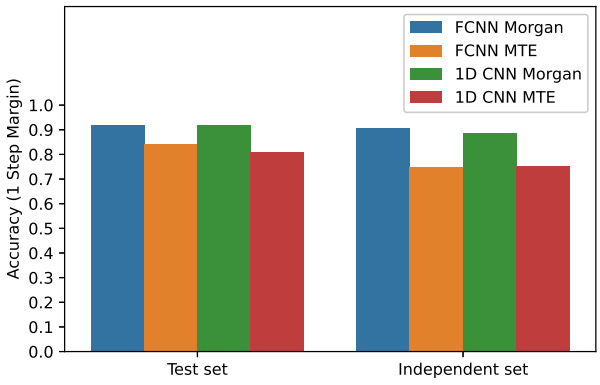

Fig. 3. Test and independent set accuracy allowing miss-classification by one step for all models.

Fig. 4 shows the main results for the regression models. Looking at these results, we can argue if we obtained better results using FCNNs or 1D CNNs. Regarding the molecular

representation, again, the use of Morgan fingerprints showed considerably better results when compared with the MTE. The FCNN with Morgan fingerprints performed better than the 1D CNN also with Morgan fingerprints with a MAE of 0.465. However, as we can see in Table VI, despite the lower MAE obtained by the FCNN with Morgan fingerprints, the 1D CNN with Morgan fingerprints obtained a slightly lower MSE and slightly higher R^2. Additionally, as shown in Fig. 4, if we check the performance of these two models in the independent set, both MAEs are similar.

TABLE VI
REGRESSION METRICS TEST SET.

Model	Features	MAE	MSE	R^2
FCNN	Morgan	0.465	0.623	0.583
FCNN	MTE	0.691	1.165	0.220
1D CNN	Morgan	0.595	0.615	0.588
1D CNN	MTE	0.737	0.888	0.405

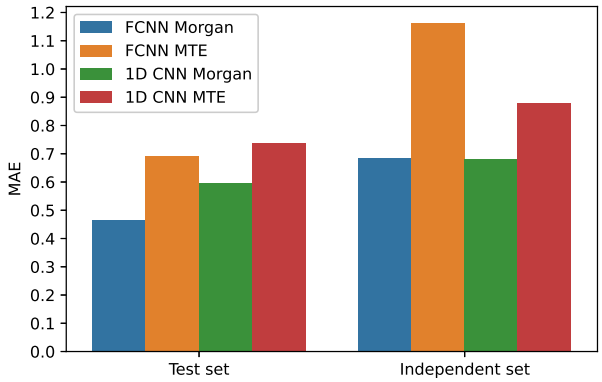

Fig. 4. Test and independent set MAE for all models.

The FCNN with Morgan fingerprints was trained, using the best parameters from the hyperparameter optimization search, for 196 epochs (early stopping with a maximum of 250 epochs) and obtained a test MAE of 0.465. This means that in mean the predictions are 0.465 steps far from the real step value.

As far as we know, there is no similar study in the literature that followed a similar approach to the one showed in this study. However, we are confident to say that the prediction of the number of biochemical transformations needed to synthesize a compound can be modeled as a DL problem. Despite the lack of other studies to compare our results with, we can say that the results obtained by our best models, a 63% accuracy, 92% if we give a one step margin, in a 5-label classification and 0.465 MAE in the regression, are promising. We also shown that, in some cases, the use of more complex molecular representations like molecular embeddings does not lead to better results. The results shown in this study corroborates that the use of traditional methods is still relevant in many cases.

IV. Conclusion

In this study, we propose the use of different DL architectures and molecular representations to predict the approximate number of biochemical transformations needed to synthesize a compound having the *E. coli* metabolites as available starting materials. The models were trained using newly generated compounds obtained by successively applying reaction rules to compounds generated in the previous step, starting by the compounds in the *E. coli* sink, up to five steps. Both Morgan fingerprints and MTE were computed and tested using FCNNs and 1D CNNs. As a result, these models showed good performance both in classification and regression tasks especially when using Morgan fingerprints and FCNNs.

As far as we know, this is the first time that the prediction of the number of biochemical transformations needed to synthesize a compound using DL is described in the literature which makes the task of comparing results hard. However, we think that the results obtained in this study indicate that approaches like this one can benefit the field of ME and specially be useful in retrobiosynthesis tools to narrow the number of generated compounds allowing the exploration of most promising pathways for the synthesis of target compounds.

In the future, it would be interesting to test other compound representations and models like recurrent neural networks and the Transformer architecture. Further exploration of the data can also be conducted to understand if the generated data are representative of what happens in microbial networks and which types of biochemical reactions are being used and left out when generating new data. Additionally, model interpretability could also be used to understand why the models make certain predictions and which properties of the molecules are more impactful for those predictions.

Acknowledgment

Centre of Biological Engineering (CEB, UMinho) for financial and equipment support. Portuguese Foundation for Science and Technology (FCT) under the scope of the strategic funding of UIDB/04469/2020 unit and through a PhD scholarship (SFRH/BD/144314/2019) awarded to João Correia.

References

[1] A. Fleming, "The discovery of penicillin," *British Medical Bulletin*, vol. 2, no. 1, pp. 4–5, 1944.

[2] M. M. El-Sheekh and H. Y. El-Kassas, "Algal production of nano-silver and gold: Their antimicrobial and cytotoxic activities: A review," *Journal of Genetic Engineering and Biotechnology*, vol. 14, no. 2, pp. 299–310, Dec. 2016.

[3] J. Du, Z. Shao, and H. Zhao, "Engineering microbial factories for synthesis of value-added products," *Journal of Industrial Microbiology & Biotechnology*, vol. 38, no. 8, pp. 873–890, Apr. 2011.

[4] S. Comba, A. Arabolaza, and H. Gramajo, "Merging engineering principles for yield improvement in microbial cell design," *Computational and Structural Biotechnology Journal*, vol. 3, no. 4, e201210016, Oct. 2012.

[5] H. Sun, H. Zhang, E. L. Ang, and H. Zhao, "Biocatalysis for the synthesis of pharmaceuticals and pharmaceutical intermediates," *Bioorganic & Medicinal Chemistry*, vol. 26, no. 7, pp. 1275–1284, Apr. 2018.

[6] S. Y. Oh, S. Y. Youn, M. S. Park, N. I. Baek, and G. E. Ji, "Synthesis of stachyobifiose using bifidobacterial -galactosidase purified from recombinant escherichia coli," *Journal of Agricultural and Food Chemistry*, vol. 66, no. 5, pp. 1184–1190, Jan. 2018.

[7] J. Zaldivar, J. Nielsen, and L. Olsson, "Fuel ethanol production from lignocellulose: A challenge for metabolic engineering and process integration," *Applied Microbiology and Biotechnology*, vol. 56, no. 1-2, pp. 17–34, Jul. 2001.

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[9] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational Intelligence and Neuroscience*, vol. 2018, pp. 1–13, 2018.

[10] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE Access*, vol. 7, pp. 19 143–19 165, 2019.

[11] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings in Bioinformatics*, bbw068, Jul. 2016.

[12] K. V. Presnell and H. S. Alper, "Systems metabolic engineering meets machine learning: A new era for data-driven metabolic engineering," *Biotechnology Journal*, vol. 14, no. 9, p. 1 800 416, May 2019.

[13] N. Hadadi and V. Hatzimanikatis, "Design of computational retrobiosynthesis tools for the design of de novo synthetic pathways," *Current Opinion in Chemical Biology*, vol. 28, pp. 99–104, Oct. 2015.

[14] B. Delépine, T. Duigou, P. Carbonell, and J.-L. Faulon, "RetroPath2.0: A retrosynthesis workflow for metabolic engineers," *Metabolic Engineering*, vol. 45, pp. 158–170, Jan. 2018.

[15] B. Liu, B. Ramsundar, P. Kawthekar, *et al.*, "Retrosynthetic reaction prediction using neural sequence-to-sequence models," *ACS Central Science*, vol. 3, no. 10, pp. 1103–1113, Sep. 2017.

[16] P. Schwaller, T. Laino, T. Gaudin, *et al.*, "Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction," *ACS Central Science*, vol. 5, no. 9, pp. 1572–1583, Aug. 2019.

[17] S. Szymkuć, E. P. Gajewska, T. Klucznik, *et al.*, "Computer-assisted synthetic planning: The end of the beginning," *Angewandte Chemie International Edition*, vol. 55, no. 20, pp. 5904–5937, Apr. 2016.

[18] H. Yim, R. Haselbeck, W. Niu, *et al.*, "Metabolic engineering of escherichia coli for direct production of 1, 4-butanediol," *Nature Chemical Biology*, vol. 7, no. 7, pp. 445–452, May 2011.

[19] P. Carbonell, P. Parutto, C. Baudier, C. Junot, and J.-L. Faulon, "Retropath: Automated pipeline for embedded metabolic circuits," *ACS Synthetic Biology*, vol. 3, no. 8, pp. 565–577, Oct. 2013.

[20] D. A. Pertusi, A. E. Stine, L. J. Broadbelt, and K. E. Tyo, "Efficient searching and annotation of metabolic networks using chemical similarity," *Bioinformatics*, vol. 31, no. 7, pp. 1016–1024, Dec. 2014.

[21] Y. Moriya, D. Shigemizu, M. Hattori, *et al.*, "PathPred: An enzyme-catalyzed metabolic pathway prediction server," *Nucleic Acids Research*, vol. 38, no. Web Server, W138–W143, Apr. 2010.

[22] T. Duigou, M. du Lac, P. Carbonell, and J.-L. Faulon, "RetroRules: A database of reaction rules for engineering biology," *Nucleic Acids Research*, vol. 47, no. D1, pp. D1229–D1235, Oct. 2018.

[23] J. G. Jeffryes, R. L. Colastani, M. Elbadawi-Sidhu, *et al.*, "MINEs: Open access databases of computationally predicted enzyme promiscuity products for untargeted metabolomics," *Journal of Cheminformatics*, vol. 7, no. 1, Aug. 2015.

[24] S. Moretti, V. D. T. Tran, F. Mehl, M. Ibberson, and M. Pagni, "MetaNetX/MNXref: Unified namespace for metabolites and biochemical reactions in the context of metabolic models," *Nucleic Acids Research*, vol. 49, no. D1, pp. D570–D574, Nov. 2020.

[25] *Daylight theory: Smarts - a language for describing molecular patterns*. [Online]. Available: https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html.

[26] G. Landrum, "Rdkit: Open-source cheminformatics software," [Online]. Available: https://github.com/rdkit/rdkit/.

[27] M. Koch, T. Duigou, and J.-L. Faulon, "Reinforcement learning for bioretrosynthesis," *ACS Synthetic Biology*, vol. 9, no. 1, pp. 157–168, Dec. 2019.

[28] J. D. Orth, T. M. Conrad, J. Na, *et al.*, "A comprehensive genome-scale reconstruction of escherichia coli metabolism—2011," *Molecular Systems Biology*, vol. 7, no. 1, p. 535, Jan. 2011.

[29] S. Gudmundsson and I. Thiele, "Computationally efficient flux variability analysis," *BMC Bioinformatics*, vol. 11, no. 1, Sep. 2010.

[30] A. Ebrahim, J. A. Lerman, B. O. Palsson, and D. R. Hyduke, "COBRApy: COnstraints-based reconstruction and analysis for python," *BMC Systems Biology*, vol. 7, no. 1, Aug. 2013.

[31] S. R. Heller, A. McNaught, I. Pletnev, S. Stein, and D. Tchekhovskoi, "InChI, the IUPAC international chemical identifier," *Journal of Cheminformatics*, vol. 7, no. 1, May 2015.

[32] *Daylight theory: Smiles*. [Online]. Available: https://www.daylight.com/dayhtml/doc/theory/theory.smiles.html.

[33] Z. Pan, Y. Chen, M. Zhou, T. A. McAllister, and L. L. Guan, "Microbial interaction-driven community differences as revealed by network analysis," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 6000–6008, 2021.

[34] A. C. Jones, K. D. Hambright, and D. A. Caron, "Ecological patterns among bacteria and microbial eukaryotes derived from network analyses in a low-salinity lake," *Microbial Ecology*, vol. 75, no. 4, pp. 917–929, Nov. 2017.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[36] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *Journal of Chemical Information and Modeling*, vol. 50, no. 5, pp. 742–754, Apr. 2010.

[37] M. Sakai, K. Nagayasu, N. Shibui, *et al.*, "Prediction of pharmacological activities from chemical structures with graph convolutional neural networks," *Scientific Reports*, vol. 11, no. 1, Jan. 2021.

[38] X. Li and D. Fourches, "Inductive transfer learning for molecular activity prediction: Next-gen QSAR models with MolPMoFiT," *Journal of Cheminformatics*, vol. 12, no. 1, Apr. 2020.

[39] M. Seo, H. K. Shin, Y. Myung, S. Hwang, and K. T. No, "Development of natural compound molecular fingerprint (NC-MFP) with the dictionary of natural products (DNP) for natural product-based drug development," *Journal of Cheminformatics*, vol. 12, no. 1, Jan. 2020.

[40] P. Morris, R. S. Clair, W. E. Hahn, and E. Barenholtz, "Predicting binding from screening assays with transformer network embeddings," *Journal of Chemical Information and Modeling*, vol. 60, no. 9, pp. 4191–4199, Jun. 2020.

[41] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017.

[42] T. R. Shultz, S. E. Fahlman, S. Craw, *et al.*, "Confusion matrix," in *Encyclopedia of Machine Learning*, Springer US, 2011, pp. 209–209.

[43] A. C. Kakas, D. Cohn, S. Dasgupta, *et al.*, "Accuracy," in *Encyclopedia of Machine Learning*, Springer US, 2011, pp. 9–10.

[44] T. Zeugmann, P. Poupart, J. Kennedy, *et al.*, "Precision," in *Encyclopedia of Machine Learning*, Springer US, 2011, pp. 780–780.

[45] M. D. Buhmann, P. Melville, V. Sindhwani, *et al.*, "Recall," in *Encyclopedia of Machine Learning*, Springer US, 2011, pp. 829–829.

[46] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2005, pp. 345–359.

[47] J. Fürnkranz, P. K. Chan, S. Craw, *et al.*, "Mean absolute error," in *Encyclopedia of Machine Learning*, Springer US, 2011, pp. 652–652.

[48] J. Fürnkranz, P. K. Chan, C. Sammut, *et al.*, "Mean squared error," in *Encyclopedia of Machine Learning*, Springer US, 2011, pp. 652–652.

[49] J. Miles, *R squared, adjusted r squared*, Sep. 2014.