Fast prediction of distances between synthetic routes with deep learning

IOP Publishing Mach. Learn.: Sci. Technol. 3 (2022) 015018 https://doi.org/10.1088/2632-2153/ac4a91

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
27 July 2021

REVISED
20 December 2021

ACCEPTED FOR PUBLICATION
12 January 2022

PUBLISHED
21 January 2022

PAPER

Fast prediction of distances between synthetic routes with deep learning

Samuel Genheden*, Ola Engkvist and Esben Bjerrum

Molecular AI, Discovery Sciences, R&D, AstraZeneca, Gothenburg, SE-431 83, Sweden
* Author to whom any correspondence should be addressed.

E-mail: samuel.genheden@astrazeneca.com

Keywords: synthetic routes, machine learning, tree edit distance, reaction informatics

Supplementary material for this article is available online

Abstract

We expand the recent work on clustering of synthetic routes and train a deep learning model to predict the distances between arbitrary routes. The model is based on a long short-term memory representation of a synthetic route and is trained as a twin network to reproduce the tree edit distance (TED) between two routes. The machine learning approach is approximately two orders of magnitude faster than the TED approach and enables clustering many more routes from a retrosynthesis route prediction. The clusters have a high degree of similarity to the clusters given by the TED-based approach and are accordingly intuitive and explainable. We provide the developed model as open-source.

1. Introduction

Machine learning (ML) and deep learning have propelled the research on computer-aided synthesis planning (CASP) in the last decade (see [1] for a recent review). CASP is becoming an integral part of drug development where models can be used to for instance predict reaction conditions or how to synthesize molecules. Retrosynthetic analysis is a technique used to find a synthetic route for a compound, in which the compound is broken down into smaller and smaller precursors until those precursors are purchasable or it is already known how to synthesize them [2, 3]. This can be formalized in a computer program, and today there exist many such algorithms (see e.g. [4–10]) and software [1] to do this. Searching for synthetic routes requires an effective search algorithm as there are typically many ways to break down a compound into smaller building blocks. A search algorithm should not only return the synthetic route that is most likely to succeed, but additionally a set of diverse routes that a chemist can analyze further if the top-ranked route is not adequate. The pruning of the identified routes to create a diverse set of routes can be done as part of the search algorithm itself [11] or as a post-processing step [12].

Recently, a clustering algorithm for predicted synthetic routes that is based on a tree edit distance (TED) calculation [13] (see figure 1 for an overview) was proposed. This is a graph theoretical metric [14, 15] of the distance between two synthetic routes that recursively utilizes chemical similarity matching. It was concluded that the TED approach produced intuitive clusters that could be easily rationalized by inspecting the reactions and molecules of the routes. Thus, the TED-based clustering approach can be used as a post-processing step to group together the results of a retrosynthetic analysis in order to aid the chemists in the analysis of the proposed routes. This algorithm could also be used to identify patent-evading routes, compare predicted routes to reported experimental routes of the same compound, or to cluster different compounds based on the similarity of their synthetic routes. Although the TED calculations produce intuitive and explainable clusters, the method is unfortunately sometimes slow and does not scale to larger sets of routes [13].

Therefore, we decided to develop a ML model that is capable of predicting the distance between two arbitrary synthetic routes. This model is similar in design to the one developed by Mo *et al* to distinguish between predicted and experimental routes [12]. Although their model can be used to cluster synthetic routes, some non-intuitive behavior of the model was observed, which was believed to be due to the training

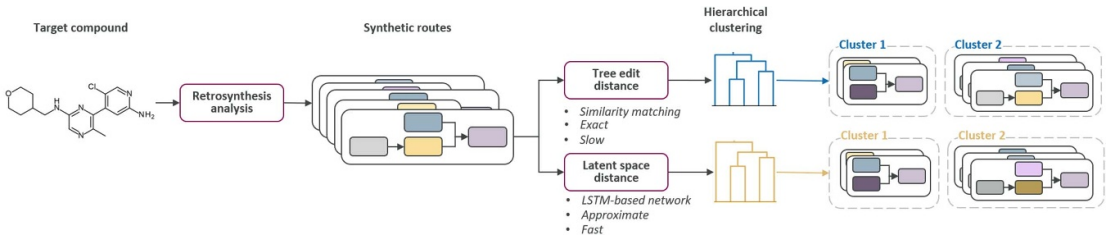

Figure 1. Overview of methods to cluster synthetic routes from a retrosynthesis analysis. The distance calculation underlying the clustering can be performed exactly but slowly using a TED calculation, or approximately but fast using an ML-model.

objective of the model. The model of Mo *et al* was never trained for the distance calculation task, thus it is not guaranteed to perform well on this task. The latent space representation of the synthetic route i.e. calculated by the model is simply a by-product of the training task. On the other hand, the TED-based approach leads to qualitative good clusters [13] and can therefore serve as the ground truth for the training of an ML model. We used a model inspired by the model of Mo *et al*, however we trained it to reproduce the TED calculations. The ML model serves as a proxy model for the expensive TED calculations. Our aim is therefore to investigate if we can develop a fast model that also leads to intuitive and explainable clusters (see figure 1). We will focus on the agreement between the TED and ML-approaches, and assume that if the ML approach recovers the TED-based clusters, the ML approach also provides intuitive and explainable clusters.

2. Methods

2.1. Compound selection and preparation

We used a set of 5000 compounds from ChEMBL [16], described and used previously for benchmarking the TED-based clustering [13]. This set will be referred to as ChEMBL-5k. We also created a new set by randomly sampling 10 000 single molecule compounds from ChEMBL that has a molecular weight between 100 and 800 D, a QED [17] score above 0.2 and is not part of the ChEMBL-5k set. The tautomeric form of the compounds was determined by RDKit [18]. This set will be referred to as ChEMBL-10k. Finally we also randomly sampled 10 000 compounds from the GDB ChEMBL and 10 000 compounds from the GDB MedChem datasets [19, 20]. The structures provided by these sets were used without further processing. These sets will be referred to as GDB-ChEMBL and GDB-MedChem, respectively.

2.2. Route predictions and TED calculations

The compounds in the four different sets were subjected to retrosynthesis predictions with the AiZynthFinder software [21]. The expansion and filter policies used by the software were derived from the USPTO dataset [22, 23], unless otherwise stated. Enamine building blocks and those available internally at AstraZeneca were used as stop criteria. The retrosynthesis search was stopped after 100 iterations, and between 5 and 25 routes were extracted, depending on the search score. For each target compound, the pairwise distance matrix of the predicted routes was calculated using a TED algorithm, as detailed previously [13].

2.3. ML model

We designed a ML model to represent a synthetic route that is based on a child-sum tree LSTM (long short-term memory) neural network model [24–26]. On each molecular node in the route we position an LSTM cell, which takes input from children nodes (i.e. precursor molecules) as well as a feature representation of the molecule. In the forward pass of the model, the latent representation of the LSTM cells are updated iteratively until the top-node (the target molecule) is reached. The feature representation of a molecule is calculated by a simple feed-forward network that takes as input a 2048 bit fingerprint (ECFP4 [27], computed by the Morgan algorithm in RDKit [18]) of the molecule. A representation of the complete network architecture is shown in figure 2. This model is similar to the model proposed by Mo *et al* [12], although we do not consider reaction fingerprints and we use a different activation function in the feed-forward network.

2.4. ML model training

We trained a twin network [28] based on the LSTM-model described above on the TED routes. Given a pair of routes, the 1st route is fed through the LSTM-model giving the latent representation of the top-node, followed by the feeding of the 2nd route, also giving a latent representation of the top-node (see figure 2). The Euclidean distance between these two latent vectors should reproduce the TED results. We used a mean

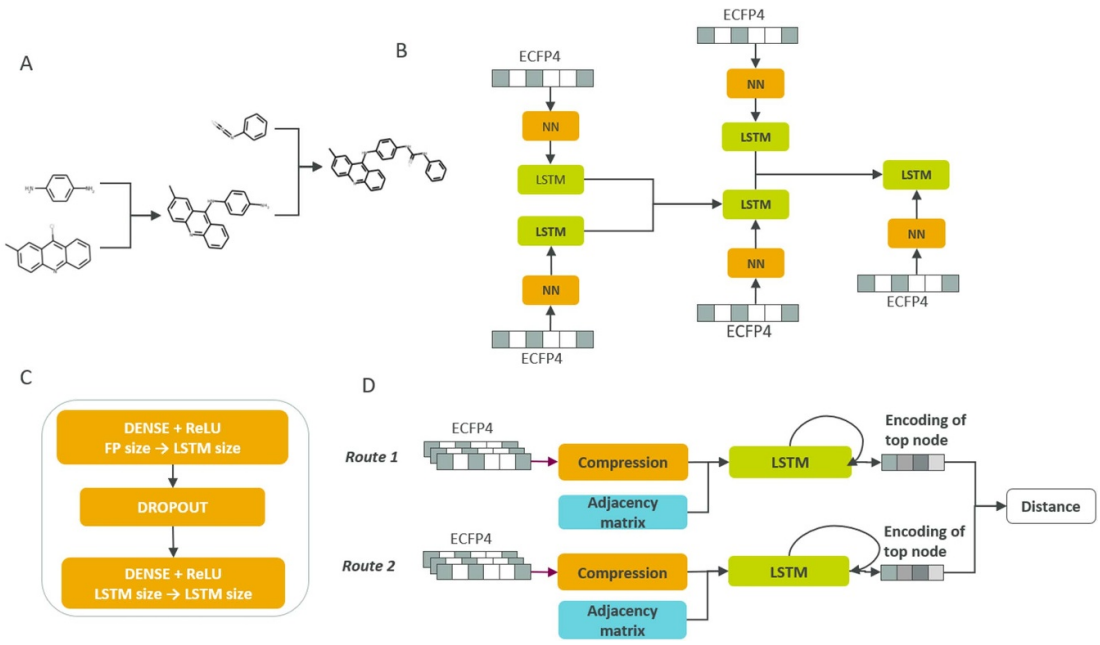

Figure 2. Illustration of the ML model. (A) A synthetic route consisting of two reactions and five molecules. (B) The conceptual representation of the route by a LSTM network. A LSTM cell is placed on top of each molecule, which takes input from the children nodes as well as a processed fingerprint of the molecule. (C) The architecture of the compression feed-forward that takes a fingerprint (with length FP size) as input and produces a shortened feature representation of the molecule (with length LSTM size). (D) The architecture of the twin network that is used to predicted distances between routes. For each route, the molecular fingerprints are fed through the compression network and the output representation is fed together with an adjacency matrix to a tree LSTM network. The adjacency matrix determines the node order in the iterative procedure that updates the LSTM cells. The LSTM encodings of the top nodes from the two routes are then used to compute an Euclidean distance.

Table 1. The total number of routes and pairs in the different sets used to train the LSTM model.

Compound set	Pair permutations	# routes	# pairs
ChEMBL-10k	Single	104 120	751 666
ChEMBL-10k	All	104 120	1399 212
ChEMBL-5k	Single	51 694	371 175
ChEMBL-5k	All	51 694	690 656

squared error-loss function together with the mean absolute error (MAE) metric to monitor the training. The Adam optimizer [29] was used with a learning rate of 0.001 and a weight decay factor of 0.001 [30]. The learning rate was decreased after a relative plateau of 10^{-4} of the loss was observed after ten epochs. The dropout probability for the feed-forward network was 0.4 and the size of the LSTM latent space was 1024. We trained in batches of 128 samples for 50 epochs. Limited hyperparameter optimization was carried out with the Optuna package [31], but because the optimization is expensive and because good performance was obtained with the above parameters, we did not attempt to fine-tune them further. The model was implemented in PyTorch [32] using the PyTorch Lightning framework [33].

The dataset consists of pairs of routes (T_i, T_j) and the calculated TED. Either both permutations of the pairs were included ($i < j$, $i = j$, $i > j$) or we only included one permutation of the pair ($i \leqslant j$). The dataset was split into approximately 80% for training, 10% for validation and 10% for testing, respectively. Care was taken so that a route is not in more than one set: we iteratively selected all pairs of routes for a random compound until we had a training set consisting of at least 80% of the total set of pairs. This procedure was repeated until we had selected at least 10% of the total set of pairs for validation and finally, the approximately 10% of pairs left were taken as the test set. As all pairs of routes originate from the same target compound, this ensures that a route is not in more than one set. Models were trained on routes generated for the ChEMBL-5k or ChEMBL-10k compound sets. The number of routes and number of pairs in the different sets is shown in table 1.

2.5. Clustering

Clustering was carried out based on the distance matrix from either TED calculations or the predictions of the ML model. We used hierarchical clustering with single linkage as implemented in SciKit-Learn [34], and

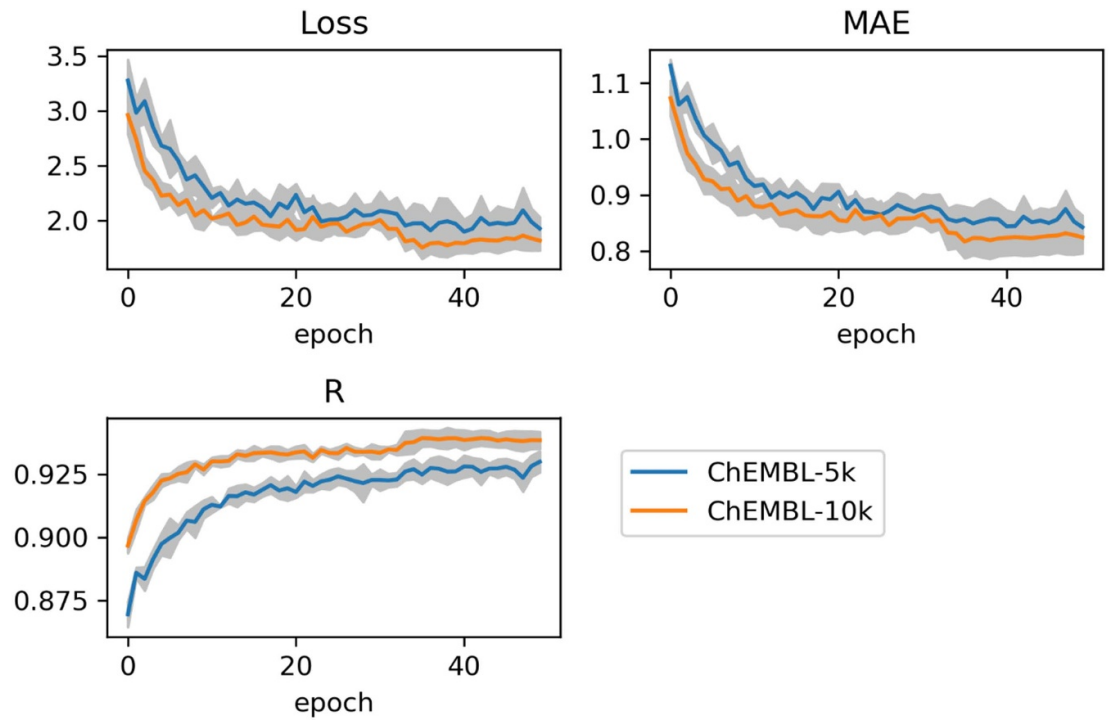

Figure 3. Average validation loss, MAE, and *R* over three independent modeling runs for ChEMBL-5k and ChEMBL-10k datasets (single pair permutation). The grey area indicate an error bar of one standard deviation. The averages and standard deviations at the final epoch are listed in table S1.

the optimal number of clusters was determined by the Silhouette method [35]. The maximum number of clusters was set to five, because the number of analyzed routes were small in order to enable comparison between the ML and TED approaches.

3. Results and discussion

3.1. The ML model learns quickly

We trained an ML model to reproduce TEDs using mainly two different datasets: single permutation of pairs for ChEMBL-5k compounds, and single permutation of pairs for ChEMBL-10k compounds. The total number of routes and pairs is listed in table 1. The average of the loss, the MAE, and *R* metrics over three independent training runs are shown in figure 3. All metrics converge after about 30 epochs, and there seems to be only a small variation between the different training runs. Therefore, we proceed with using the models produced from the first training runs when evaluating them further below. In table S1 (available online at stacks.iop.org/MLST/3/015018/mmedia), we compare the MAE and *R* when using a single permutation of the pairs to using all permutations of the pairs. For both MAE and *R*, the models trained on all pairs perform slightly worse than the models trained on only single permutations, although it is unclear if the differences are significant due to the low number of independent training runs. Therefore, we continued evaluation on the models trained on single pair permutations, because we can speed up the training time by only including about half the number of pairs. As the twin network share weights for the two LSTM models and because the Euclidean distance calculation is commutative, it should not matter which route is passed first through the network, which furthers motivated us to continue with the single-permutation training sets. Finally, there appears to be no significant difference in performance when comparing the ChEMBL-5k and ChEMBL-10k models.

3.2. ML-based clustering is significantly faster than TED-based clustering

The ML model trained on the ChEMBL-10k dataset (single pair permutation) was used to predict distances between routes for the ChEMBL-5k, GDB-MedChem, and GDB-ChEMBL compound sets. For the ChEMBL-10k compound sets, the model trained on the ChEMBL-5k dataset was used. The foremost reason to develop a proxy model such as the ML model is that the TED calculations were observed to be too slow in the worst case and did not scale well. Therefore, it is of interest to compare the timings of the two approaches. The mean and worst clustering time (distance + clustering calculations) is shown in table 2. If

Table 2. Timings in seconds for TED- and ML-based clustering for four datasets.

		TED		ML	
	Mean # pairs	Mean time	Worst time	Mean time	Worst time
ChEMBL-5k	63.90	6.45	213.95	0.05	0.70
ChEMBL-10k	64.75	5.54	211.89	0.05	1.41
GDB-MedChem	50.76	6.71	156.83	0.06	2.24
GDB-ChEMBL	52.97	6.82	157.30	0.06	2.16
ChEMBL-5k (100 routes)	4770.86	377.03	4215.92	0.72	5.62
ChEMBL-5k (all solved routes)	1217.81	72.62	3806.78	0.20	4.65

Table 3. Metrics show the similarity of distances and clusters when comparing the TED and ML approaches for various datasets. The ML model was trained on the ChEMBL-10k or ChEMBL-5k sets.

		Distance		Cluster	
Compound set	Expansion policy	*R*	MAE	Mean similarity	Median similarity
ChEMBL-5k	USPTO	0.95	0.92	0.88	0.97
ChEMBL-10k	USPTO	0.95	1.03	0.87	0.95
GDB-MedChem	USPTO	0.92	1.66	0.87	0.96
GDB-ChEMBL	USPTO	0.92	1.61	0.87	0.95
GDB-MedChem	Reaxys	0.92	1.55	0.88	1.00
GDB-ChEMBL	Reaxys	0.92	1.53	0.88	1.00

we only look at the cases where we extract a small number of routes (1st four rows), the mean time for TED calculation is between approximately 6 and 7 s, compared to 0.1 for the ML approach. The worst time is between 157 and 214 s for the TED approach and only between 0.7 and 2 s for the ML approach. This shows that both on average and in worst-case scenario the ML approach is more than an order of magnitude faster than the TED calculations, and potentially two orders faster. If we instead extract 100 routes for each compound, we see that the speed-up is even more impressive. The mean and worst time is only 0.7 and 6 s, respectively for the ML method, which is negligible compared to the search time. In addition to extracting 100 routes, we also extract all solved routes for compounds where a solution could be found, and between 5 and 25 routes for compounds where a solution could not be found. These timings are shown in table 2 as well, and we can conclude that the average and worst clustering time is acceptable for this scenario as well. Therefore, we can conclude that we have successfully created a faster proxy model.

3.3. ML-based clustering recovers the TED clusters to a large extent

In table 3 we show the MAE and correlation coefficient, *R* when comparing the TED with the distances predicted by the ML model. For the ChEMBL-5k set, *R* is 0.95 and the MAE is 1 distance unit, which shows that there is a strong correlation and only a small deviation. This can also be seen in figure 4; there are a few distances that are considerably different, especially distances where the ML predictions is much lower than the TED. We see similar *R* and MAE if we compare the distances for the routes of the ChEMBL-10k set, showing that a model trained a smaller dataset (Chembl-5k) is as good as training on a larger dataset (Chembl-10k). Having such a good prediction of distances, it is not surprising that the clustering performance is good as well. The mean similarity is between 0.87 and 0.88, with the median being between 0.95 and 0.97. This shows that most of the routes that are clustered together with TED are also clustered together with the ML approach. We can thus conclude that the ML approach at least on average produces satisfactory clusters. In the supporting information, we investigate the differences between the TED and ML approaches for different subsets of the data (see tables S2 and S3). This analysis show that the discrepancy between the approaches will be larger when the routes to compare contains more reactions and if the routes are converged. However, the cluster similarity is expected to be reasonably consistent.

3.4. ML model for route distances and clustering is transferable

We included compounds from GDB-MedChem and GDB-ChEMBL in the study to investigate the transferability power of the model. The GDB database contains enumerated compounds that in general are harder to break down with the AiZynthFinder tool (see table S4). This shows that the compounds in GDB contain chemistry that is not well-represented in the USPTO dataset, on which the expansion policy was trained. And even if one of the GDB sets were created to be similar to ChEMBL compounds [20], the

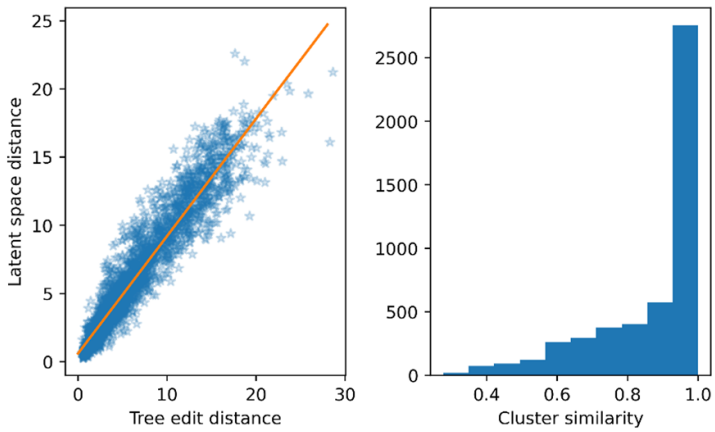

Figure 4. Performance of the ML model trained on the ChEMBL-10k dataset and applied on routes for ChEMBL-5k compounds. (a) Correlation between TED and ML prediction of distances. The scatterplot is a random subset of 1% of all the pairwise distances whereas the solid line show a linear regression line for all the data. The correlation coefficient, *R*, for this regression line is 0.95. (b) The distribution of cluster similarities for all compounds. The average and median for this distribution is listed in table 3.

compounds in this set are clearly much harder to break down than the compounds ChEMBL-10k and ChEMBL-5k sets, indicating that the compounds are different at least from a synthesizability perspective. However, as we see in table 3 the performance of the ML approach on the GDB sets is very satisfactory. The correlation coefficient is slightly weaker than for the ChEMBL sets, and the MAE points at a larger discrepancy. Still, the mean and median cluster similarities for the GDB sets are on a par with the similarities seen for the ChEMBL sets, showing that the clustering algorithm is robust to distance discrepancies. This finding can be understood by the fact that we use the same approach to generate routes for ChEMBL and GDB compounds, which would lead to the same type of reaction chemistry being encoded in the routes. Thus the relation between molecules in the route is not inherently different when comparing ChEMBL and GDB routes, and therefore the ML model can successfully transfer the knowledge from routes for ChEMBL compounds to routes for GDB compounds. We also predicted routes for the GDB sets using an expansion policy trained on the Reaxys database [22, 36], and we can see in table S4 that this expansion policy is more successful than the USPTO-based expansion policy in finding routes for the GDB compounds. We still reproduce the TED calculations when we predict route distances and clusters using the ML model trained in ChEMBL-10k: as seen in table 3, the *R* and MAE when comparing the distances are 0.92 and approximately 1.5, respectively, and the mean cluster similarity is 0.88 for both GDB sets. Thus, we can be relatively certain that the trained ML model is predictive even if one changes the expansion policy or have compounds that are not ChEMBL-like.

3.5. Large clustering differences between TED-based and ML-based clustering can be rationalized

We investigate the difference in clustering further by inspecting the distances and clusters for the compound where the cluster similarity was the lowest (0.28) when analyzing the Chembl-5k set. For this compound, the optimal number of clusters according to the Silhouette method for the TED matrix is 5, whereas for the latent space distance matrix it is 2. Representative routes for each cluster is shown in figure S1. The difference in the number of clusters can be understood from the dendrograms of the two distance matrices shown in figure 5. For the ML predictions, route number 4 is so distant from the other routes that it is placed into a singleton cluster, and all the other routes in a second cluster. For the TED approach, the distances are more evenly distributed and it is therefore more preferential to form more clusters. Still, the routes clustered first in the hierarchical clustering would be very similar for the two approaches. For instance, routes 0 and 1 would be clustered early and then joined with route 3 in both TED- and ML-based clustering. Similarly, the cluster consisting of routes 5 and 9 would be joined with the cluster formed from routes 6 and 10. That route 4 is different to the other routes can be rationalized by realizing that this route is the only one that ends with a dehydration reaction (see figures 6 and S1). So the ML model single out this feature much more than the TED approach. TED clustering places route 4 closer to routes 0–3, which are single-step routes similar to the 1st step of route 4. This analysis highlights that the TED and ML approaches sometimes focus on different chemistry in the analyzed routes. Similar analyses for two other compounds are shown in figures S2 and S3, concluding that the two approaches sometimes give different clusters but the clusters can for both approach be rationalized. It should be pointed out the optimal number of clusters has a subjective element to it, so that

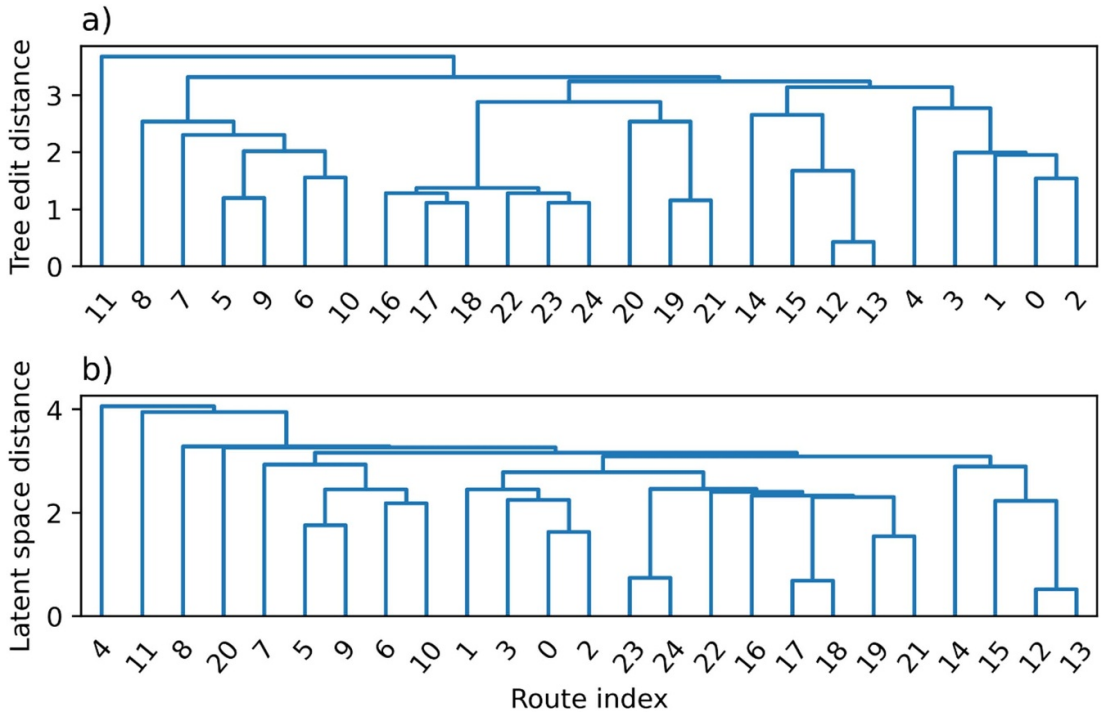

Figure 5. Dendrograms formed from distances matrixes from (a) TED calculations, (b) ML predictions when performing hierarchical clustering. The routes are for the target compound where the cluster agreement between the TED and ML approaches is lowest as discussed in the main text.

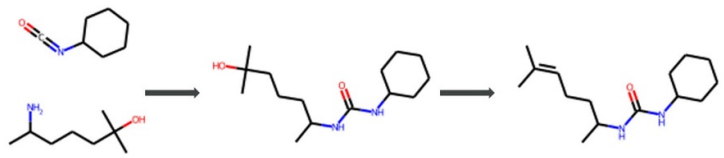

Figure 6. Route number 4 from the dendrograms in figure 5. It is the only route ending with a dehydration reaction.

the two methods gives different number of clusters is not inherently wrong. For 58% of the compounds in the Chembl-5k set, the optimal number of clusters according to the Silhouette method is equal when comparing the TED and ML approaches, for 24% of the compounds the ML approach leads to more clusters and in 19% the TED approach leads to more clusters.

4. Conclusions

A novel method to rapidly compute the similarity between synthetic routes has been developed and can be used to, e.g. cluster results from a retrosynthesis analysis, cluster compounds based on route similarity, compare predicted and experimentally reported routes, or in the comparison of synthesis planning tools. It would also be of interest to investigate if it could be utilized directly in the Monte Carlo tree search to guide the search into novel areas of the search space. We showed that the novel method can reproduce the synthetic route clustering based on TEDs. Because the cluster similarity between the ML and TED approaches is high, the ML model will provide chemists with intuitive and useful clusters. Furthermore, we showed that the novel method is fast: on average the predictions take less than one second and in the worst-case a few seconds, which is negligible compared to the route prediction time. Finally, we also showed that the trained ML model is transferable: is robust to changes in targeted chemical space and the expansion policy used in the retrosynthesis analysis. We thus envisage that the novel method presented herein will be useful in synthesis planning. The developed method is provided as open-source and is available in the AiZynthFinder software [21].

Data availability statement

The code for AiZynthFinder is available: https://github.com/MolecularAI/aizynthfinder and the code for the route distance calculations: https://github.com/MolecularAI/route-distances.

The data that support the findings of this study are openly available at the following URL/DOI: https://zenodo.org/record/4925903.

Acknowledgments

Amol Thakkar is acknowledged for proof-reading the text.

ORCID iDs

Samuel Genheden https://orcid.org/0000-0002-7624-7363
Esben Bjerrum https://orcid.org/0000-0003-1614-7376

References

[1] Johansson S, Thakkar A, Kogej T, Bjerrum E, Genheden S, Bastys T, Kannas C, Schliep A, Chen H and Engkvist O 2020 AI-assisted synthesis prediction *Drug Discov. Today Technol.* **32–33** 65–72
[2] Corey E J and Todd Wipke W 1969 Computer-assisted design of complex organic syntheses *Science* **166** 178–92
[3] Coley C W, Green W H and Jensen K F 2018 Machine learning in computer-aided synthesis planning *Acc. Chem. Res.* **51** 1281–9
[4] Heifets A and Jurisica I 2012 Construction of new medicines via game proof search *Proc. National Conf. Artificial Intelligence* pp 1564–70
[5] Segler M H S, Preuss M and Waller M P 2018 Planning chemical syntheses with deep neural networks and symbolic AI *Nature* **555** 604–10
[6] Klucznik T *et al* 2018 Efficient syntheses of diverse, medicinally relevant targets planned by computer and executed in the laboratory *Chemistry* **4** 522–32
[7] Coley C W *et al* 2019 A robotic platform for flow synthesis of organic compounds informed by AI planning *Science* **365** eaax1566
[8] Lin K, Xu Y, Pei J and Lai L 2020 Automatic retrosynthetic route planning using template-free models *Chem. Sci.* **11** 3355–64
[9] Schwaller P, Petraglia R, Zullo V, Nair V H, Haeuselmann R A, Pisoni R, Bekas C, Anna I and Laino T 2020 Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy *Chem. Sci.* **11** 3316–25
[10] Chen B, Li C, Dai H and Song L 2020 Retro*: learning retrosynthetic planning with neural guided A* search (arXiv:2006.15820)
[11] Shibukawa R, Ishida S, Yoshizoe K, Wasa K, Takasu K, Okuno Y, Terayama K and Tsuda K 2020 CompRet: a comprehensive recommendation framework for chemical synthesis planning with algorithmic enumeration *J. Cheminform.* **12** 52
[12] Mo Y, Guan Y, Verma P, Guo J, Fortunato M E, Lu Z, Coley C W and Jensen K F 2021 Evaluating and clustering retrosynthesis pathways with learned strategy *Chem. Sci.* **12** 1469–78
[13] Genheden S, Engkvist O and Bjerrum E 2020 Clustering of synthetic routes using tree edit distance *ChemRxiv* (https://doi.org/10.26434/chemrxiv.13372475.v1)
[14] Pawlik M and Augsten N 2015 Efficient computation of the tree edit distance *ACM Trans. Database Syst.* **40** 1–40
[15] Pawlik M and Augsten N 2016 Tree edit distance: robust and memory-efficient *Inf. Syst.* **56** 157–73
[16] Gaulton A *et al* 2012 ChEMBL: a large-scale bioactivity database for drug discovery *Nucleic Acids Res.* **40** D1100
[17] Bickerton G R, Paolini G V, Besnard J, Muresan S and Hopkins A L 2012 Quantifying the chemical beauty of drugs *Nat. Chem.* **4** 90–98
[18] Landrum G RDKit: open-source cheminformatics (available at: www.rdkit.org)
[19] Awale M, Sirockin F, Stiefl N and Reymond J-L 2019 Medicinal chemistry aware database GDBMedChem *Mol. Inform.* **38** 1900031
[20] Bühlmann S and Reymond J L 2020 ChEMBL-likeness score and database GDBChEMBL *Front. Chem.* **8** 46
[21] Genheden S, Thakkar A, Chadimová V, Reymond J L, Engkvist O and Bjerrum E 2020 AiZynthFinder: a fast, robust and flexible open-source software for retrosynthetic planning *J. Cheminform.* **12** 70
[22] Thakkar A, Kogej T, Reymond J L, Engkvist O and Bjerrum E J 2020 Datasets and their influence on the development of computer assisted synthesis planning tools in the pharmaceutical domain *Chem. Sci.* **11** 154–68
[23] Genheden S, Engkvist O and Bjerrum E J 2020 A quick policy to filter reactions based on feasibility in AI-guided retrosynthetic planning *ChemRxiv*
[24] Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80
[25] Tai K S, Socher R and Manning C D 2015 Improved semantic representations from tree-structured long short-term memory networks *53rd Annu. Meet. Assoc. Comput. Linguist. 7th Int. Jt. Conf. Nat. Lang. Process. Asian Fed. Nat. Lang. Process. Proc. Conf.* vol 1 pp 1556–66
[26] Dawe J *pytorch-tree-lstm* (available at: https://github.com/unbounce/pytorch-tree-lstm)
[27] Rogers D and Hahn M 2010 Extended-connectivity fingerprints *J. Chem. Inf. Model.* **50** 742–54
[28] Chicco D 2021 Siamese neural networks: an overview *Artificial Neural Networks* vol 2190 Methods in Molecular Biology (New York: Humana) pp 73–94
[29] Kingma D P and Ba J L 2015 Adam: a method for stochastic optimization *3rd Int. Conf. on Learning Representations, ICLR 2015—Conf. Track Proc.* (International Conference on Learning Representations, ICLR)
[30] Loshchilov I and Hutter F 2017 Decoupled weight decay regularization (arXiv:1711.05101)
[31] Akiba T, Sano S, Yanase T, Ohta T and Koyama M 2019 Optuna: a next-generation hyperparameter optimization framework *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min* pp 2623–31
[32] Paszke A *et al* 2019 PyTorch: an imperative style, high-performance deep learning library (arXiv:1912.01703)
[33] Falcon W *et al* 2020 PyTorchLightning/pytorch-lightning: 0.7.6 release (available at: https://github.com/PyTorchLightning/pytorch-lightning)

[34] Pedregosa F *et al* 2011 Scikit-learn: machine learning in Python *J. Machine Learning Res.* **12** 2825–30
[35] Rousseeuw P J 1987 Silhouettes: a graphical aid to the interpretation and validation of cluster analysis *J. Comput. Appl. Math.* **20** 53–65